natural

Version 4.1.2 for Mainframes | Editors

SOFTWARE AG

This document applies to Natural Version 4.1.2 for Mainframes and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Table of Contents

Editors - Overview 1
Editors - Overview 1
Editors - General Information 2
Editors - General Information 2
Object Names 3
Split-Screen Mode 4
Split-Screen Commands 4
Editor Profile 6
General Information 6
Additional Options 8
Editor Defaults 9
General Defaults 10
Color Definitions 11
Direct Commands 12
User Exit USR0070P 13
Exit Profile Maintenance 13
Tutorial - Using the Map Editor 14
Tutorial - Using the Map Editor 14
Components of the Map Editor 15
Invoking the Map Editor 16
Session 1 - Designing a Map, Line and Field Commands 17
Session 2 - Processing Rules 31
Session 3 - Extended Field Editing 37
Session 4 - INPUT USING MAP 43
Session 5 - WRITE USING MAP, Fields from a View 45
Program Editor 51
Program Editor 51
Invoking the Program Editor 51
Top Information Line 52
Bottom Information Line 52
Editor Command Line 52
Editing a Program 52
Multiple Functions 52
Dynamic Conversion from Lower to Upper Case 53
Editor Commands 54
Editor Commands for Positioning 57
Line Commands 58
Special PF-Key Functions 60
Cursor-Sensitive Commands 61
The SCAN Commands 61
The SPLIT Command 61
The EDIT and LIST System Commands 62
The Exit Function 62
Data Area Editor 63
Data Area Editor 63
Invoking the Data Area Editor 63
Top Information Line 64
Bottom Information Line 64
Editor Command Line 64
Editing a Data Area 65
Editor Commands 67
Line Commands 69
Extended Field Definition Editing 72

Exit Function . . . 75
Sharing Data Area Sources by Natural Version 3.1 and 4.1 . . . 75
Map Editor . . . 77
Map Editor . . . 77
Components of the Map Editor . . . 78
Summary of Map Creation . . . 79
Step 1 . . . 79
Step 2 . . . 79
Step 3 . . . 79
Step 4 . . . 79
Invoking the Map Editor . . . 80
Overview of Functions . . . 82
Initializing a Map . . . 86
Delimiters . . . 87
Format . . . 88
Context . . . 90
Filler Characters . . . 91
Editing a Map . . . 92
Commands and Function Keys for Positioning . . . 93
Line Commands . . . 94
Field Commands . . . 96
Defining Map Fields . . . 98
Defining Fields Directly on the Screen . . . 98
Selecting Fields from a User View or Data Definition . . . 99
Using System Variables in a Map Definition . . . 100
Extended Field Editing . . . 101
HE Parameter Syntax: . . . 105
Post Assignment Function . . . 106
Array and Table Definition . . . 107
Array Definition . . . 107
Table Definition . . . 111
Processing Rules . . . 115
Field-Related Processing Rules . . . 115
Function-Key-Related Processing Rules . . . 116
Processing Rule Editing . . . 117

Editors - Overview

This document provides an overview of the Natural editors and a short summary of their functions.

The following topics are covered:

- Editors - General Information

  An overview of which Natural objects are edited with which Natural editor. In addition, it contains information on Natural object names, split-screen mode and the editor profile.

- Program Editor

  You use the program editor to write and maintain Natural programs, subprograms, subroutines, classes, copycode, helproutines and text elements.

- Data Area Editor

  Three different types of data areas can be created and edited with the data area editor:

  - **Local data areas** (LDAs)
    LDAs are used to define the data to be used within a single Natural program.
  - **Global data areas** (GDAs)
    GDAs are used to define the data to be used by one or more Natural programs.
  - **Parameter data areas** (PDAs)
    PDAs are used to specify the data parameters to be passed between a Natural program and a subprogram, external subroutine, helproutine, dialog or method.

- Map Editor

  You use the map editor to create and edit maps. After a map is created, you can store it in a library and invoke it using an INPUT USING MAP statement.

- Tutorial - Using the Map Editor

  This section contains a series of tutorial sessions which introduce you to the use of the Natural map editor.

- DDM Editor (SYSDDM Utility)

  The editor used for Natural data definition modules (DDMs) is provided with the SYSDDM utility.

  The SYSDDM Utility is described in the Natural Utilities documentation.

- Software AG Editor

  The Software AG Editor is used to edit objects in the following Software AG products: Natural for Mainframes, Natural ISPF, Entire Operations, Entire Output Management.

  The Software AG Editor is described in the relevant documentation.

Editors - General Information

This section gives an overview of which Natural objects are edited with which Natural editor. In addition, it contains information on Natural object names, split-screen mode and the editor profile.

You invoke a Natural editor with the system command EDIT as described in the Natural System Command Reference documentation.

Which editor is invoked depends on the type of object you wish to edit:

- Programs, subprograms, subroutines, helproutines, copycode and text are created and edited in the program editor.
- Global data areas, local data areas and parameter data areas are created and edited in the data area editor.
- Maps and help maps are created and edited in the map editor.
- Predict descriptions are edited in the Predict description editor
  (see the Predict documentation).

An online help system is provided with each editor.

Tutorials which introduce you to the main features of the editors are provided in the documentation Natural for Mainframes - First Steps and in the section Tutorial - Using the Map Editor.

In addition to the Natural editors, the Software AG Editor is used by several Natural utilities and other Software AG products (for further information, see the relevant section in the Natural Installation Guide for Mainframes and the Software AG Editor documentation).

> **Note:**
> If you wish to use the Software AG Editor as an alternative to the Natural program editor, Natural ISPF must be installed.

This section covers the following topics:

- Object Names
- Split-Screen Mode
- Editor Profile

Object Names

The name of a Natural object can be 1 to 8 characters long. It can consist of the following characters:

Character	Explanation
A - Z	upper-case alphabetical characters
0 - 9	numeric characters
-	hyphen
_	underline
/	slash
$	dollar sign
&	ampersand (only as language code character; see also the section Defining the Language of a Natural Object)
#	hash/number sign
+	plus sign (only allowed as first character)

The first character of the name must be one of the following:

- an upper-case alphabetical character
- #
- +

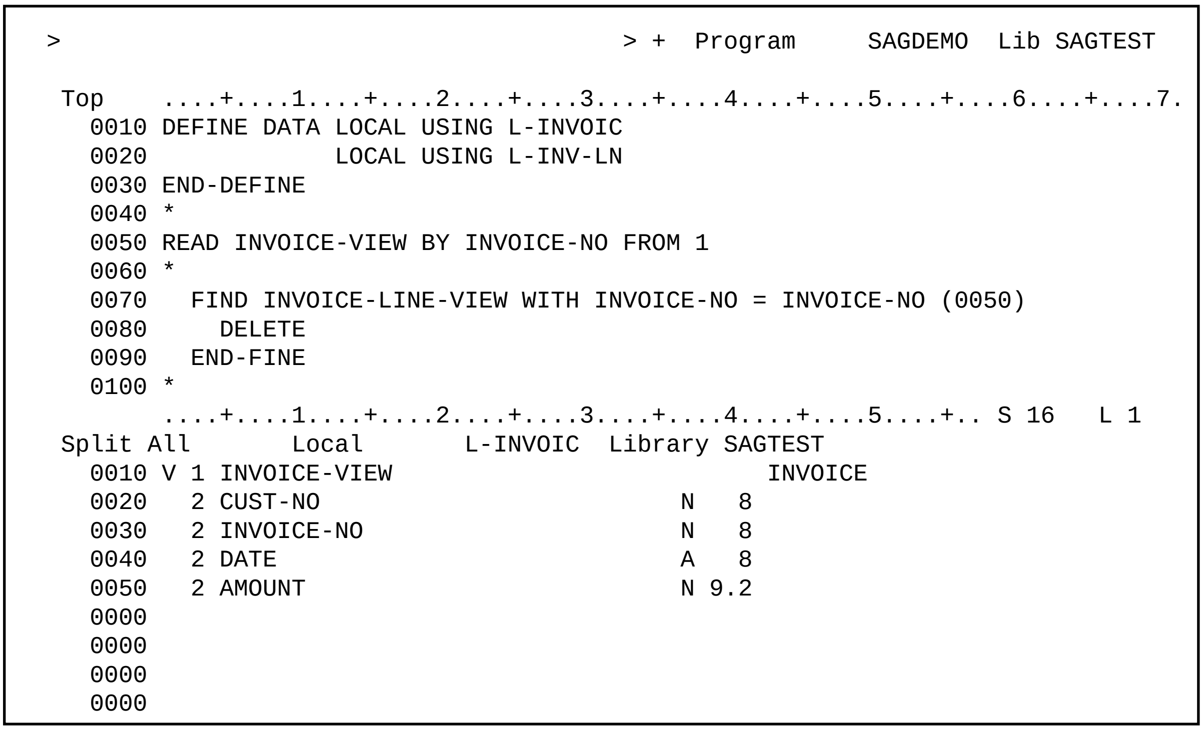
If the first character is a hash/number (#) sign or a plus (+) sign, the name must consist of at least one additional character.

Split-Screen Mode

You can use all three Natural editors in split-screen mode: you can use one half of the screen for editing an object and at the same time have another Natural object displayed in the other half. Split-screen mode can be used to display a view, a data area, a Predict program description or a Natural program in the lower half of the screen. In addition, you can include items shown in the display section of the screen into the editing section that is, into the object you are currently editing.

Example:

The following figure shows the program editor in split-screen mode with the source code of a program in the editing section (upper half) and a local data area in the display section (lower half):



```
>                                       > +  Program     SAGDEMO  Lib SAGTEST

 Top    ....+....1....+....2....+....3....+....4....+....5....+....6....+....7.
   0010 DEFINE DATA LOCAL USING L-INVOIC
   0020             LOCAL USING L-INV-LN
   0030 END-DEFINE
   0040 *
   0050 READ INVOICE-VIEW BY INVOICE-NO FROM 1
   0060 *
   0070   FIND INVOICE-LINE-VIEW WITH INVOICE-NO = INVOICE-NO (0050)
   0080     DELETE
   0090   END-FINE
   0100 *
        ....+....1....+....2....+....3....+....4....+....5....+.. S 16   L 1
 Split All       Local       L-INVOIC  Library SAGTEST
   0010 V 1 INVOICE-VIEW                          INVOICE
   0020   2 CUST-NO                     N   8
   0030   2 INVOICE-NO                  N   8
   0040   2 DATE                        A   8
   0050   2 AMOUNT                      N 9.2
   0000
   0000
   0000
   0000
```

Split-Screen Commands

The following commands can be used to display and position an object in split-screen mode. All commands begin with an **S** or with SPLIT to indicate the working mode - Split Screen. The SPLIT command is a cursor-sensitive command as described in the section Program Editor.

Command	**Function**
S ++	Position to bottom of object.
S B	
S - -	Position to top of object.
S T	
S +	Position one page forwards.
S +P	
S -	Position one page backwards.
S -P	
S +*nnn*	Position *nnn* lines forwards (only valid for program editor).
S -*nnn*	Position *nnn* lines backwards (only valid for program editor).
S .	Terminate split-screen mode.
S END	
S DATA *name* [*library*]	Display data area (global, local, parameter).
S DESCRIPTION *pgm-name* [*library*]	Display program description (if available) from the Predict Data Dictionary (valid for program and data area editor only).
S FUNCTION *name* [*library*]	Display the subroutine *name*, where *name* is the name of the subroutine as used in the DEFINE SUBROUTINE statement (not the name of the object containing the subroutine). This command is only available in the program editor.
S PROGRAM *name* [*library*]	Display program, subprogram, subroutine, helproutine, copycode, text, map, class.
S SCAN [*value*]	Scan for a *value*. Each line containing the value is marked with a greater than (>) sign. To further scan for the same value, enter **S SC** only.
S VIEW *name* [SHORT]	Display view (DDM, as defined in Predict or SYSDDM). If SHORT is specified, the DDM is listed in short form (that is, only the Adabas short names and corresponding Natural field names are displayed) without any field header or field edit mask information.

In the data area editor, with DATA, PROGRAM and VIEW, an asterisk (*) can be used for *name* to display a list of all available objects. If the asterisk (*) is preceded by one or more characters, only those objects whose names begin with these characters are displayed.

A *library* can be specified with the program editor only. Under Natural Security, a library cannot be specified.

Editor Profile

This section covers the following topics:

- General Information
- Additional Options
- Editor Defaults
- General Defaults
- Color Definitions
- Direct Commands
- User Exit USR0070P
- Exit Profile Maintenance

General Information

When working with the Natural program editor or data area editor, an editor profile can be defined per user.

The editor profile shows the functions assigned to the PF and PA keys, and various other settings to be in effect during the edit session.

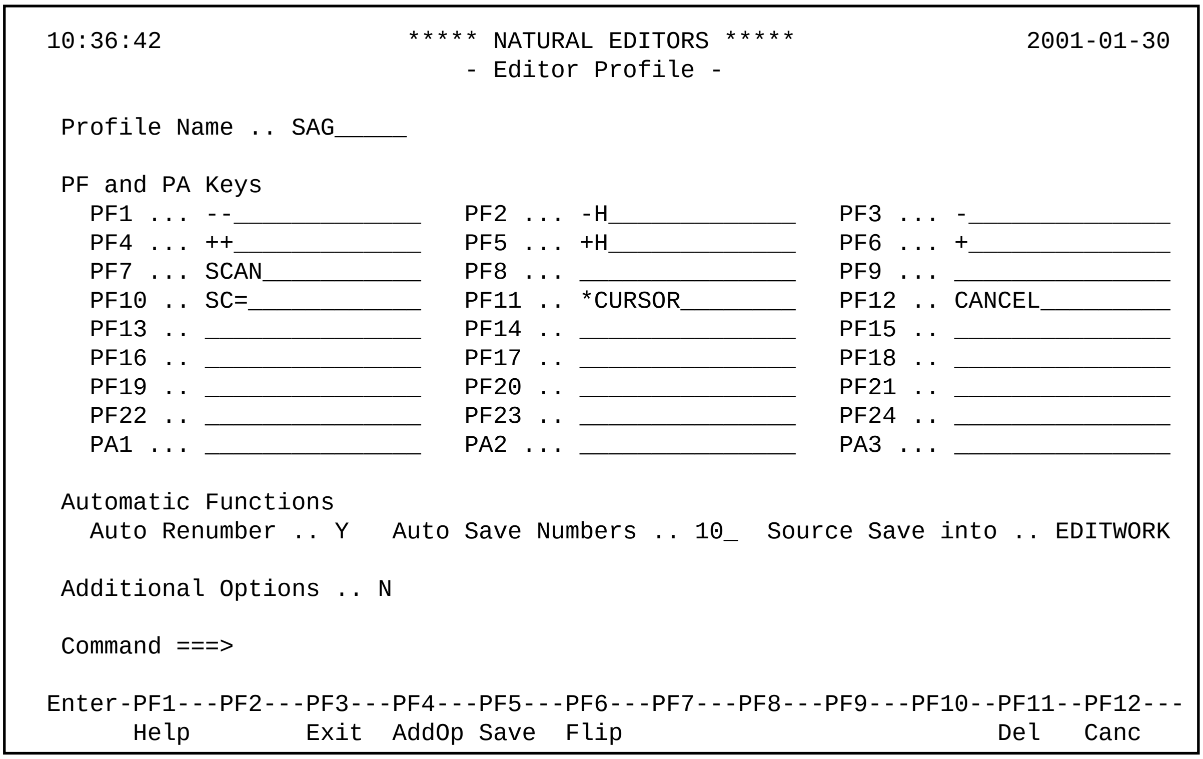
The profile can be modified by the users to suit their personal editing requirements.

To display your current profile, enter the command PROFILE in the command line of your program or data area editor. If such a profile does not exist, the default profile SYSTEM is displayed which can be used to create a user’s profile. The SYSTEM profile is read from the user exit USR0070P and can be modified there.

To display the profile of another user or the default profile SYSTEM, enter the command PROFILE *profile-name*, where *profile-name* corresponds to the respective user ID.

When you are in an edit session and enter the PROFILE command together with your own user ID as profile name, your profile is always invoked directly from the database; any modifications made during the current session, but not yet saved on the database, will not apply. Therefore, to invoke your current session profile, enter the PROFILE command only.

When you enter the PROFILE command, the following screen is displayed:



```
10:36:42                 ***** NATURAL EDITORS *****                2001-01-30
                             - Editor Profile -

 Profile Name .. SAG_____

 PF and PA Keys
   PF1 ... --_____________   PF2 ... -H_____________   PF3 ... -______________
   PF4 ... ++_____________   PF5 ... +H_____________   PF6 ... +______________
   PF7 ... SCAN___________   PF8 ... _______________   PF9 ... _______________
   PF10 .. SC=____________   PF11 .. *CURSOR________   PF12 .. CANCEL_________
   PF13 .. _______________   PF14 .. _______________   PF15 .. _______________
   PF16 .. _______________   PF17 .. _______________   PF18 .. _______________
   PF19 .. _______________   PF20 .. _______________   PF21 .. _______________
   PF22 .. _______________   PF23 .. _______________   PF24 .. _______________
   PA1 ... _______________   PA2 ... _______________   PA3 ... _______________

 Automatic Functions
   Auto Renumber .. Y   Auto Save Numbers .. 10_  Source Save into .. EDITWORK

 Additional Options .. N

 Command ===>

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help        Exit  AddOp Save  Flip                          Del   Canc
```

Attention:
Profile modifications made during the current session are lost when you enter the system command LOGON.

Entry	Explanation
Profile Name	The name of the editor profile. Your own editor profile is displayed. If such a profile does not exist, you can modify the default profile to suit your own requirements. To do so, overtype the profile name SYSTEM with your user ID and save the renamed profile on the database. If you overtype the name of your profile with any other valid profile name (that is, any other valid user ID) and press ENTER, the profile of the corresponding user is invoked. Only one profile can be established per user ID, and any modifications made to another user’s profile are only valid for the current session; they cannot be saved on the database. You can, however, overtype the profile name of another user’s profile with your own user ID and then save the renamed profile on the database.
PF and PA Keys	The commands assigned to the PF and PA keys are displayed. Any Natural editor or system command can be assigned. Combinations of commands (separated by a comma) are also possible.
Auto Renumber	**Y** indicates that the source code in the program editor is to be renumbered automatically if any of the following occurs: ● a CATALOG, CHECK, RUN, SAVE or STOW command is issued; ● a **.I** line command is issued and no line number is available for the line to be inserted. **Note:** See also Renumbering of Source-Code Line Number References.
Auto Save Numbers	If a numeric value is entered, a copy of the source is saved automatically into the member specified in the "Source Save into" field after the specified number of modifications have taken place. Modification means each time that the source has been changed as a result of information entered on the screen. Auto Save Numbers applies to the map editor, too.
Source Save into	The name of the member into which a copy of the source is to be saved automatically; the default name EDITWORK can be modified. The specified member is overwritten each time the number of changes specified in the "Auto Save Numbers" field has been exceeded.

Additional Options

If you mark Additional Options on the Editor Profile screen with **Y** or press PF4, a window will be displayed from which you can select the following options:

- Editor Defaults
- General Defaults
- Color Definitions

A plus (+) sign in front of an option indicates that some values have already been set in the corresponding window or via an appropriate editor command.

To select an option, you mark it with a **Y**.

For each option selected, a corresponding window will be displayed. The individual items of each window are explained below.

Editor Defaults

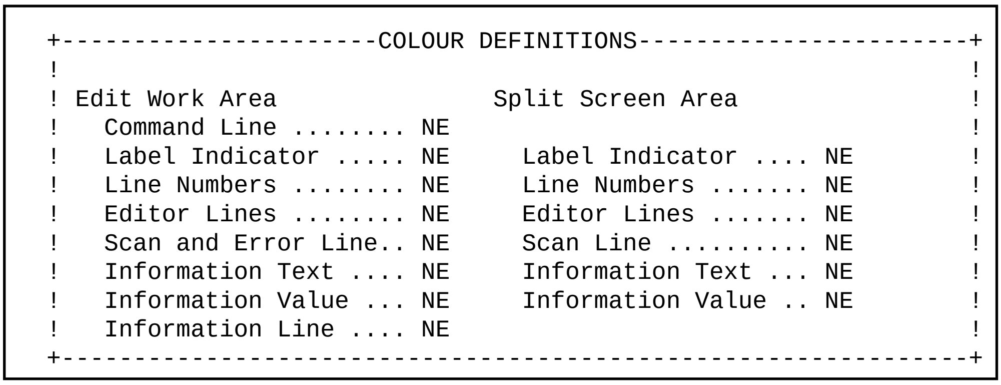
Option	Explanation
Escape Character for Line Command	The escape character which must precede each line command; the default escape character is a period (.).
Empty Line Suppression	Y Any lines left blank are eliminated from the source as soon as you press ENTER. N Any lines left blank are **not** eliminated from the source when you press ENTER. This parameter only applies to the line command **.I** (see the sections Program Editor and Data Area Editor).
Source Size Information	Y The actual size of the object being edited and the remaining space available is displayed in the bottom information line of the editor screen. In addition, in the program editor, the programming mode (reporting or structured) is displayed in the top information line of the editor screen. N No such information is displayed.
Source Status Message	Y A transaction message will be displayed in the top information line each time the source is modified, checked, saved, cataloged or stowed. N No such transaction message will be displayed. The Source Status Message parameter only applies to the program editor.
Absolute Mode for SCAN/CHANGE	Y Corresponds to the editor command SET ABS ON. N Corresponds to the editor command SET ABS OFF. See Editor Commands in the sections Program Editor and Data Area Editor.
Range Mode for SCAN/CHANGE	Y Corresponds to the editor command SET RANGE ON. N Corresponds to the editor command SET RANGE OFF. See Editor Commands in the section Program Editor.
Direction Indicator	Indicates the direction (**+** or **-**) in which several editor commands are to work (see also Editor Command Line in the sections Program Editor and Data Area Editor).

General Defaults

Parameter	**Explanation**
Editing in Lower Case	Y Lower-case characters in the source code are **not** automatically converted to upper case (corresponds to the terminal command %L). N Lower-case characters in the source code are automatically converted to upper case (corresponds to the terminal command %U). Automatic conversion is in effect by default.
Dynamic Conversion of Lower Case	This option is relevant only if the above option is set to **Y**: Y All lower-case characters in the source code are automatically converted to upper case - except text strings that are enclosed in apostrophes and comments: these remain as you enter them (see also the section Program Editor). N Any source code remains as you enter it.
Position of Message Line	Indicates the position of the message line; possible values are **TOP**, **BOT**, ***nn*** and ***-nn***.
Cursor Position in Command Line	Y Indicates that the cursor is positioned in the edit command line after the source has been modified and you pressed ENTER.
Stay on Current Screen	Y Corresponds to the editor command SET STAY ON. N Corresponds to the editor command SET STAY OFF. See Editor Commands in the sections Program Editor and Data Area Editor.
Prompt Window for Exit Function	Y When you enter the EXIT command in the editor command line, a confirmation window is displayed (see also Exit Function in the sections Program Editor and Data Area Editor).
ISPF Editor as Program Editor	Y Natural ISPF (if installed) is invoked instead of the Natural program editor.

Color Definitions

If you mark Color Definitions with **Y** in the Additional Options window, the following window will be displayed:



```
+----------------------COLOUR DEFINITIONS-----------------------+
!                                                               !
! Edit Work Area                Split Screen Area               !
!   Command Line ........ NE                                    !
!   Label Indicator ..... NE      Label Indicator .... NE       !
!   Line Numbers ........ NE      Line Numbers ....... NE       !
!   Editor Lines ........ NE      Editor Lines ....... NE       !
!   Scan and Error Line.. NE      Scan Line .......... NE       !
!   Information Text .... NE      Information Text ... NE       !
!   Information Value ... NE      Information Value .. NE       !
!   Information Line .... NE                                    !
+---------------------------------------------------------------+
```

In this window you can specify the colors in which the various parts of the edit-work and split-screen area of your program or data area editor are to be displayed.

To get a list of the colors available, you enter the question mark (?) help character in any of the input fields of the Color Definitions window or press PF1 (Help).

Apart from the Command and Information Lines and the corresponding Information Text and Values, the following individual parts can be assigned a specific color:

Label Indicator	Leftmost column of the editor screen; used, for example, to label a source code line on which a certain command has been performed (for example, the **.X** and **.Y** line commands).
Line Numbers	Column of the source code line numbers (program editor only).
Editor Lines	Lines of source code currently in the edit-work and/or split-screen area.
Scan and Error Line	All lines marked with an **S** (or a greater than (>) sign in split-screen mode) as a result of a scan operation, any line where an error was detected (marked with **E** and applicable in edit-work area of program editor only) and the error message line itself.

Direct Commands

The following direct commands can be used instead of the corresponding PF keys. Direct commands have to be entered in the command line at the bottom of the editor profile screen.

Command	Description
CANCEL	This command (or PF12) cancels the current function and returns you to the screen from which it was invoked. Any modifications made to the profile have no effect for the current session.
DELETE	This command (or PF11) deletes the current profile from the database. Before the profile is deleted, however, a confirmation window pops up, in which you can either type the name of the profile and press ENTER to confirm the deletion of the profile, or press ENTER only to exit the function.
EXIT	This command (or PF3) invokes the exit function prompt window, regardless of whether the corresponding editor default parameter (see General Defaults) is set or not.
FLIP	This command (or PF6 and PF18) is used to switch between the two PF-key lines.
REFRESH	This command (or PF13) displays the profile parameters currently valid for the session, which means that any modifications made so far, but not yet saved, are overwritten.
SAVE	This command (or PF5) saves all currently valid profile parameters both for the current session and on the database. However, it does **not** leave the current function.

User Exit USR0070P

The user exit routine USR0070P enables you to modify the parameter settings in the default profile SYSTEM. USR0070P provides a list of all parameters which are to receive a default setting.

With this user exit, you can also determine whether editor profiles are to be stored in the FNAT system file, the FUSER system file or the scratch-pad file.

Exit Profile Maintenance

To exit from any editor profile maintenance function, press PF3 (Exit) or enter the EXIT command in the command line at the bottom of your terminal screen. In both cases the EXIT Function prompt window is invoked offering you the following options:

Function	Explanation
Save and Exit	Returns you to the screen from where the current profile maintenance function was invoked and saves any modifications made to the current profile. Modifications are saved both for the current session and on the database. If you are working with another user’s editor profile, however, modifications made to that profile cannot be saved on the database. They are valid for the current session only; a corresponding message is returned.
Exit without Saving	Returns you to the screen from where the current profile maintenance function was invoked. Any modifications made to the current profile are only valid for the current session; they are **not** saved on the database. Pressing ENTER corresponds to "Exit without Saving".
Resume Function	Closes the prompt window and returns you to the current profile maintenance function.

Tutorial - Using the Map Editor

This tutorial is not intended to be a comprehensive description of the full range of possibilities provided by the Natural map editor. Rather, these sessions represent a general introduction to how the map editor may be used. Therefore, explanations are kept to a minimum. For a full description of all functions and features, see the section Map Editor.

It this important that you work through the sessions in the sequence below.

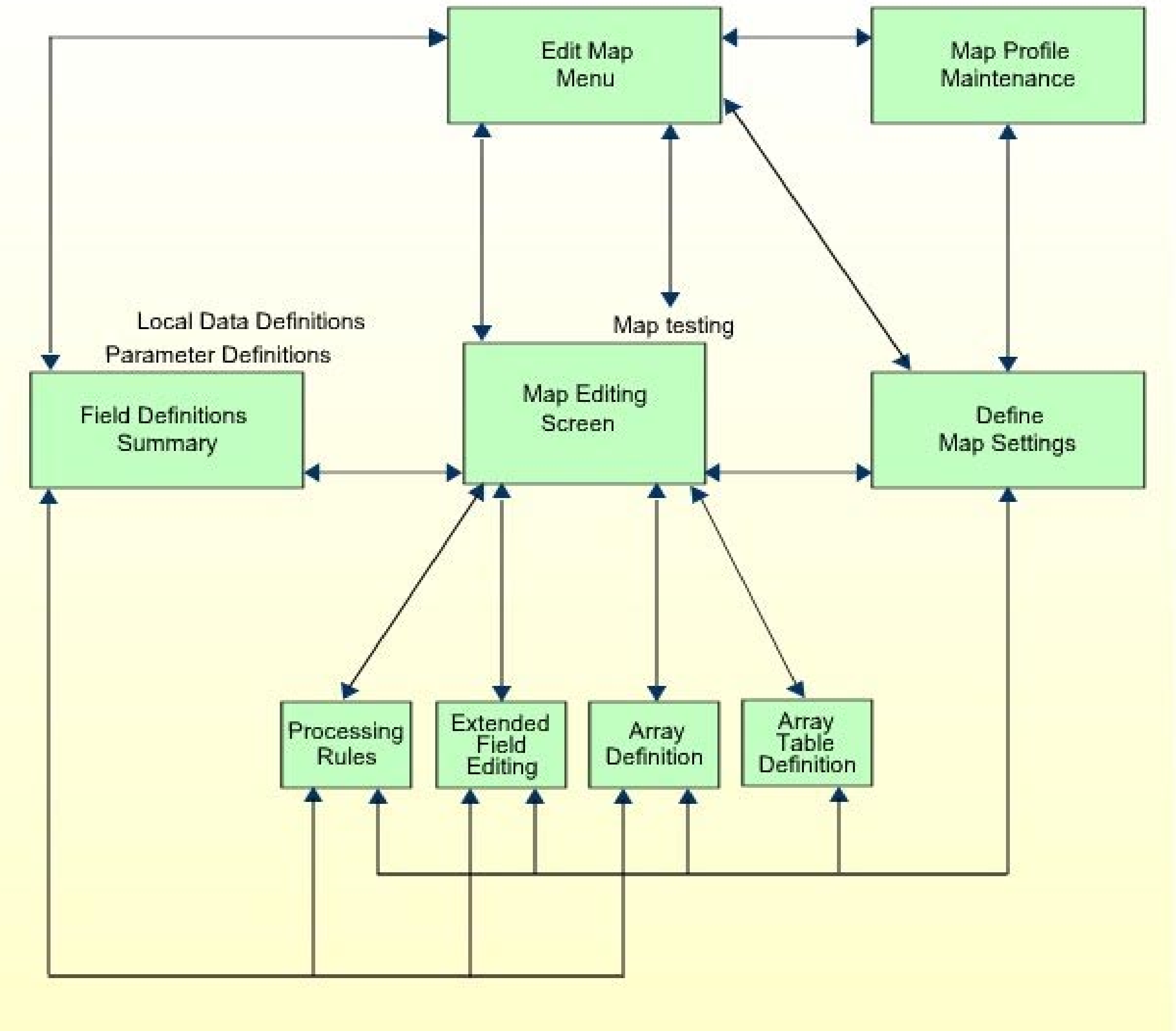
The Natural map editor is used to create *maps* (screen layouts) referenced in a program. The map editor allows *direct manipulation of fields* for screen layouts; *extended field editing* facilitates the definition of fields; *processing rules* within the map can be associated with these fields. Once a map has been created, it may be stored in the Natural system file from where it may be invoked by a Natural program using a WRITE USING MAP or INPUT USING MAP statement.

This section covers the following topics:

- Components of the Map Editor
- Invoking the Map Editor
- Session 1 - Designing a Map, Line and Field Commands
- Session 2 - Processing Rules
- Session 3 - Extended Field Editing
- Session 4 - INPUT USING MAP
- Session 5 - WRITE USING MAP, Fields from a View

Components of the Map Editor

The following diagram gives an overview of the various sections of the map editor.

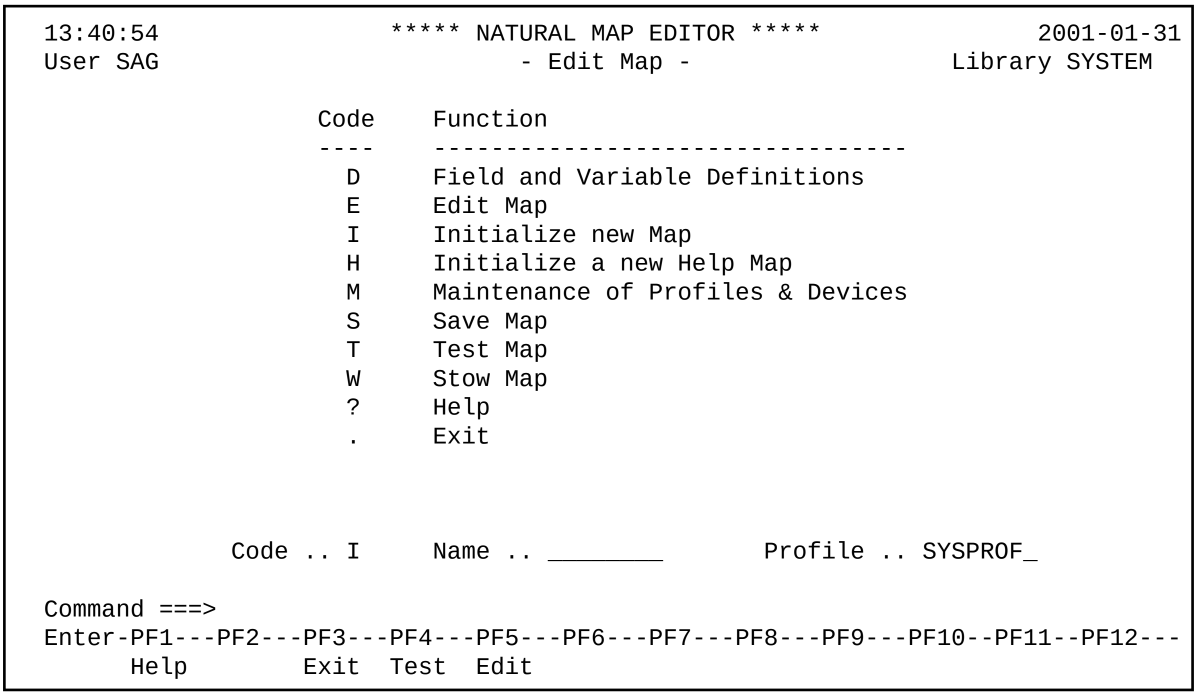


Invoking the Map Editor

On the Natural Main Menu, select "Development Functions". The Development Functions menu will be displayed.

For the sessions in this tutorial, change the mode to "Structured" if you are not working in structured mode already. (If you are currently working in reporting mode, enter an "S" in the first position of the Mode field.)

Then enter an "E" (Edit) in the Code field and an "M" (Map) in the Type field. The Edit Map menu will be displayed:



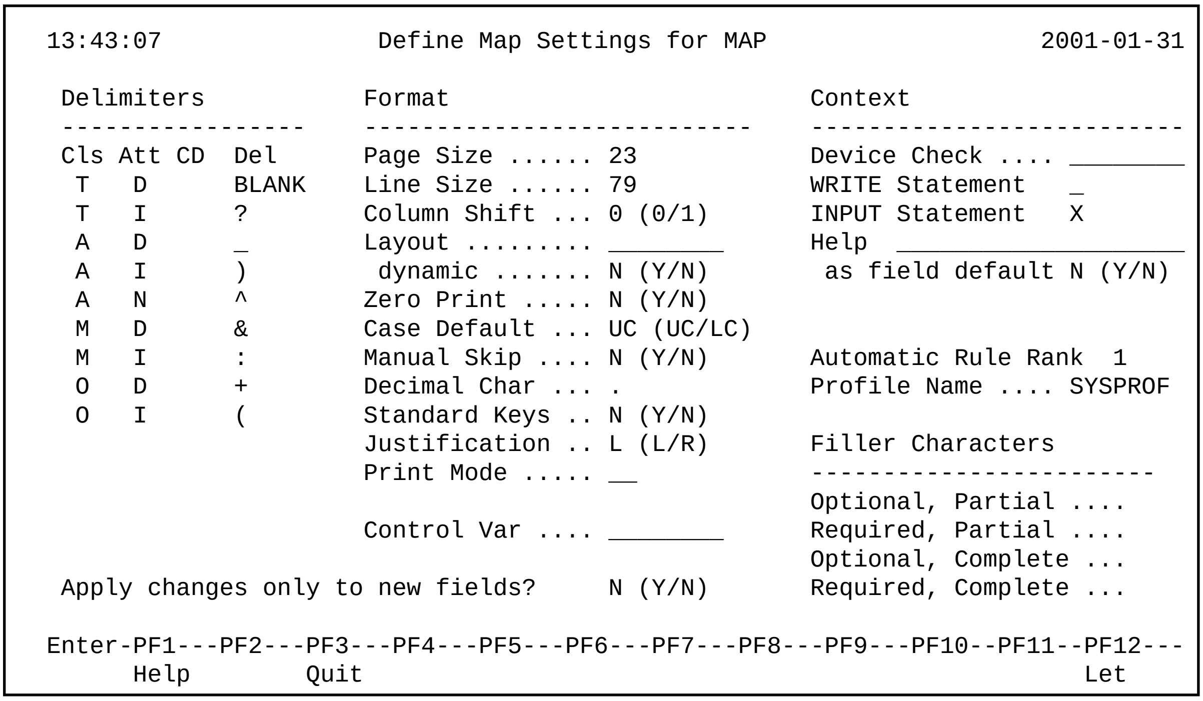
```
 13:40:54                ***** NATURAL MAP EDITOR *****             2001-01-31
 User SAG                          - Edit Map -                   Library SYSTEM

                   Code    Function
                   ----    ---------------------------------
                     D     Field and Variable Definitions
                     E     Edit Map
                     I     Initialize new Map
                     H     Initialize a new Help Map
                     M     Maintenance of Profiles & Devices
                     S     Save Map
                     T     Test Map
                     W     Stow Map
                     ?     Help
                     .     Exit



             Code .. I     Name .. ________       Profile .. SYSPROF_

 Command ===>
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help        Exit  Test  Edit
```

The Edit Map menu is the main menu of the Natural map editor.

Note:
The map editor contains an extensive help system. Anytime you require help, enter a question mark (?) in the field for which you wish further information. This will invoke the online help for that field. If a field does not have an individual help assigned, a help menu will be displayed, from which you may select the desired item of information.

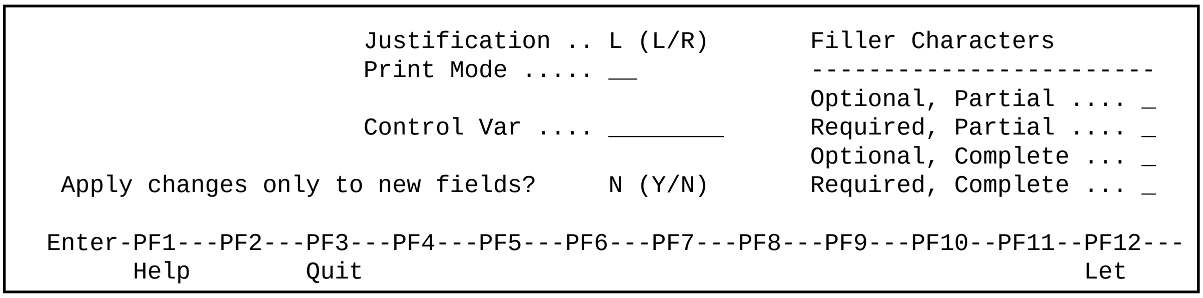
Session 1 - Designing a Map, Line and Field Commands

On the Edit Map menu, enter an "I" (Initialize new Map) in the Code field and MAP001 in the Name field. The Define Map Settings For Map screen will be invoked:



```
13:43:07               Define Map Settings for MAP                   2001-01-31

 Delimiters           Format                       Context
 -----------------    ---------------------------    --------------------------
 Cls Att CD  Del      Page Size ...... 23            Device Check .... ________
  T   D      BLANK    Line Size ...... 79            WRITE Statement   _
  T   I      ?        Column Shift ... 0 (0/1)       INPUT Statement   X
  A   D      _        Layout ......... ________      Help  ____________________
  A   I      )         dynamic ....... N (Y/N)        as field default N (Y/N)
  A   N      ^        Zero Print ..... N (Y/N)
  M   D      &        Case Default ... UC (UC/LC)
  M   I      :        Manual Skip .... N (Y/N)       Automatic Rule Rank  1
  O   D      +        Decimal Char ... .             Profile Name .... SYSPROF
  O   I      (        Standard Keys .. N (Y/N)
                      Justification .. L (L/R)       Filler Characters
                      Print Mode ..... __            ------------------------
                                                     Optional, Partial ....
                      Control Var .... ________      Required, Partial ....
                                                     Optional, Complete ...
 Apply changes only to new fields?     N (Y/N)       Required, Complete ...

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help        Quit                                                  Let
```

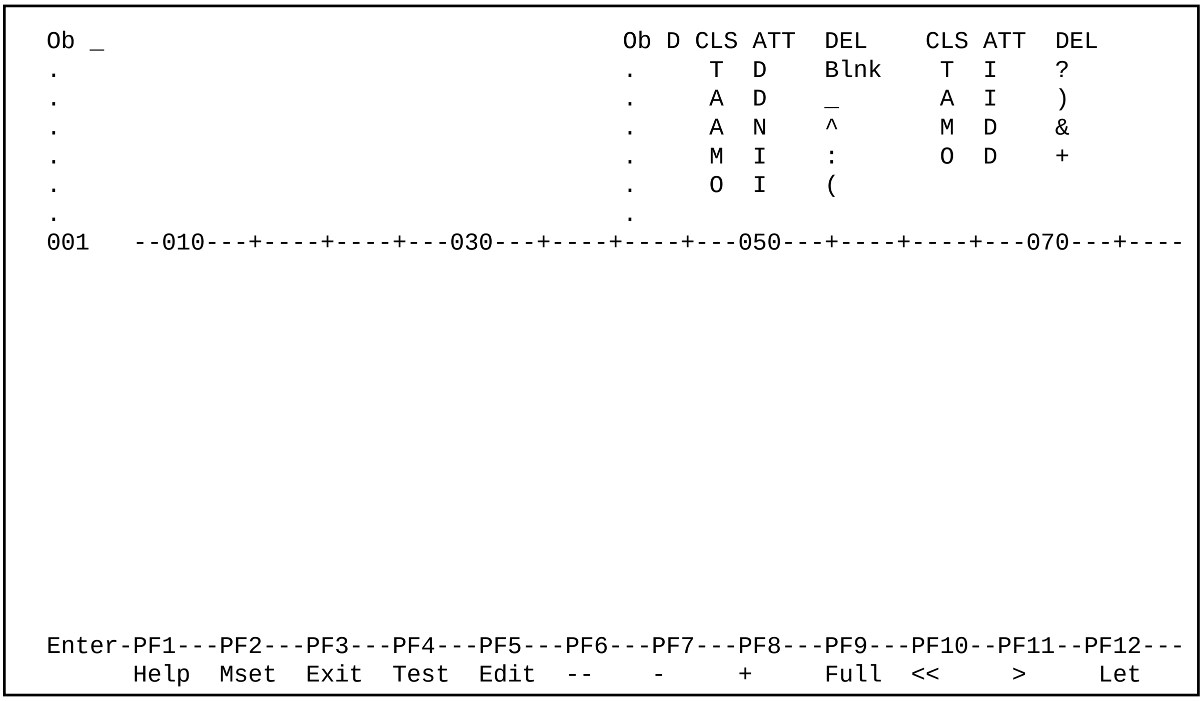
Move the cursor to the "Filler Characters" section of the screen. Type in an underscore (_) after each of the four options as shown below:



```
                        Justification .. L (L/R)        Filler Characters
                        Print Mode ..... __             ------------------------
                                                        Optional, Partial .... _
                        Control Var .... ________       Required, Partial .... _
                                                        Optional, Complete ... _
   Apply changes only to new fields?      N (Y/N)        Required, Complete ... _

  Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
        Help        Quit                                                   Let
```

This will cause any empty positions within an input field on the map to be filled with the underscore (_). This enables the user to see the exact position and length of a field, which makes entering input easier.

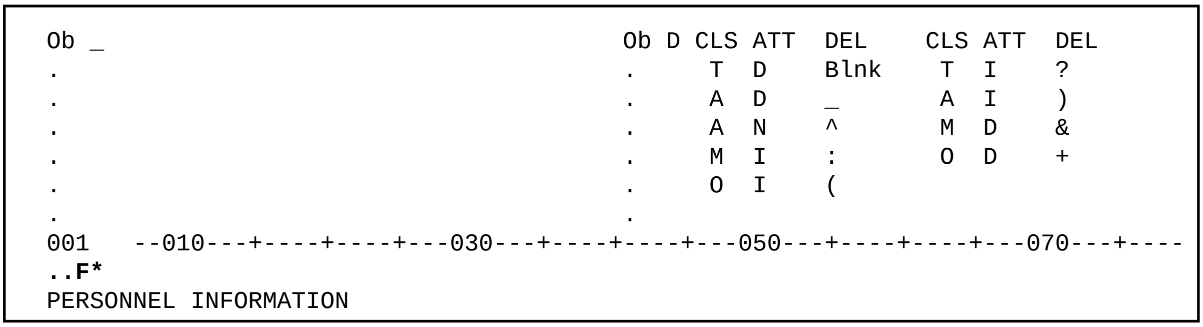
Ignore the other map settings for the time being and press ENTER. Press ENTER again. The map editing screen will be invoked:



```
  Ob _                                      Ob D CLS ATT  DEL    CLS ATT  DEL
  .                                         .     T  D    Blnk    T  I    ?
  .                                         .     A  D    _       A  I    )
  .                                         .     A  N    ^       M  D    &
  .                                         .     M  I    :       O  D    +
  .                                         .     O  I    (
  .                                         .
  001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----














  Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
        Help  Mset  Exit  Test  Edit  --    -     +     Full  <<     >     Let
```

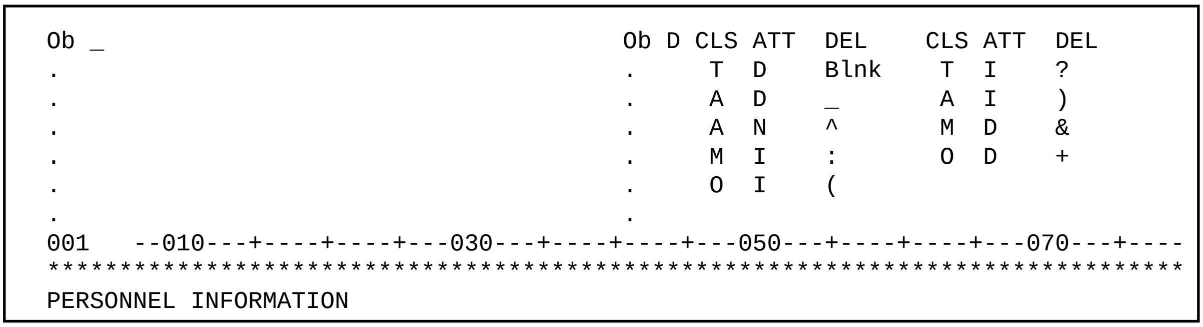
The map editing screen will appear in split-screen format, the top half displaying the delimiter characters which are valid for the map to be created, while the bottom half is the area where you actually design a map.

In the first line of the editing area, enter the line command "..F*", and in the second line type in the text PERSONNEL INFORMATION as shown below:



```
Ob _                                    Ob D CLS ATT  DEL    CLS ATT  DEL
.                                       .     T  D    Blnk    T  I    ?
.                                       .     A  D    _       A  I    )
.                                       .     A  N    ^       M  D    &
.                                       .     M  I    :       O  D    +
.                                       .     O  I    (
.                                       .
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
..F*
PERSONNEL INFORMATION
```

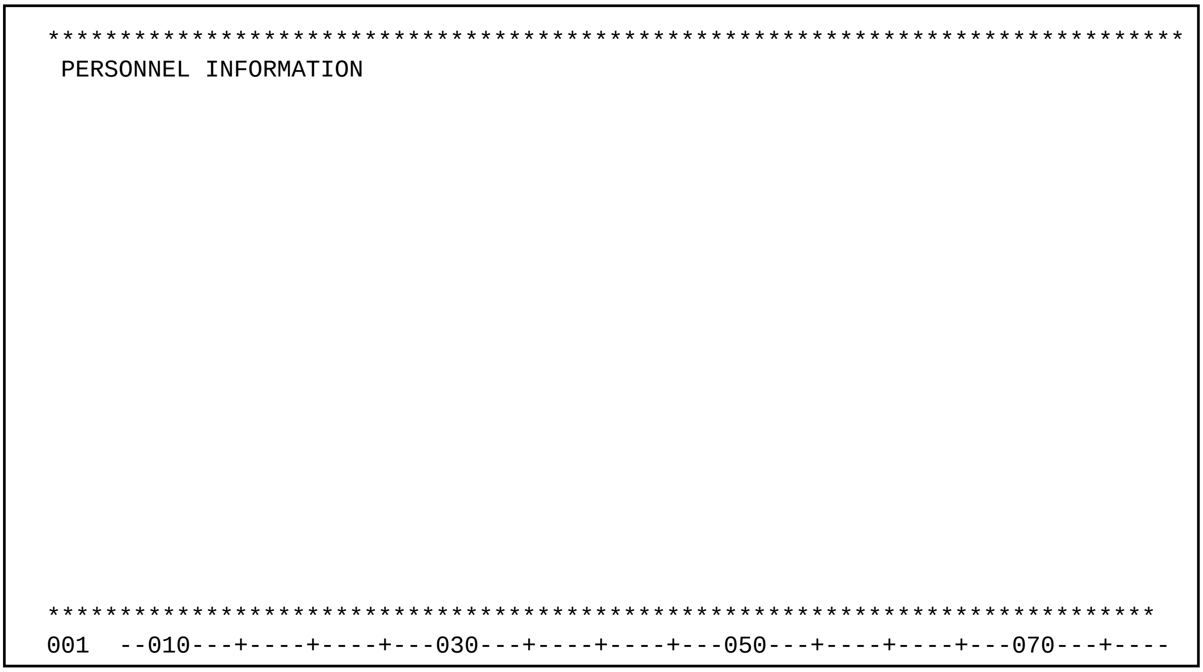
The result will be as follows:



```
Ob _                                    Ob D CLS ATT  DEL    CLS ATT  DEL
.                                       .     T  D    Blnk    T  I    ?
.                                       .     A  D    _       A  I    )
.                                       .     A  N    ^       M  D    &
.                                       .     M  I    :       O  D    +
.                                       .     O  I    (
.                                       .
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
********************************************************************************
PERSONNEL INFORMATION
```

Press PF9 to obtain the full-screen map editing area.

In the bottom line, enter the line command "..F*". The screen now appears as follows:



```
********************************************************************************
 PERSONNEL INFORMATION

















******************************************************************************
001  --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

Type in the line command "..C" in the first three positions of the text:

```
*******************************************************************************
 ..CSONNEL INFORMATION
```

As a result, the text will be centered.

Enter the following as shown on the screen below:

```
*******************************************************************************
(*DATX                       PERSONNEL INFORMATION
(*TIMX

PLEASE ENTER CITY::X(20)
PLEASE ENTER NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

*DATX and *TIMX are Natural system variables which will display the current date and time respectively. The opening parenthesis "(" is the delimiter for intensified output fields. The colon (:) is the delimiter for intensified modifiable fields. The number of Xs indicate the length of the fields.

The map will appear as follows:

```
*******************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE ENTER CITY::XXXXXXXXXXXXXXXXXXXX
PLEASE ENTER NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

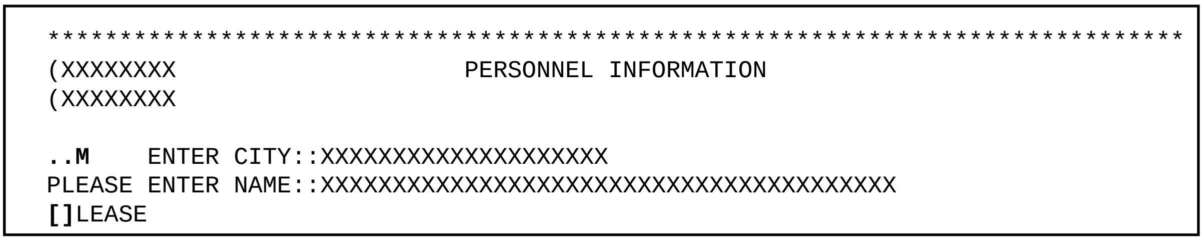
Type in the field command ".M" and move the cursor to the position indicated by **[]** as shown below:

```
*******************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

.MEASE ENTER CITY::XXXXXXXXXXXXXXXXXXXX
PLEASE ENTER NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
[]
```

When you press ENTER, the text field where the command was entered will be moved to the cursor position:

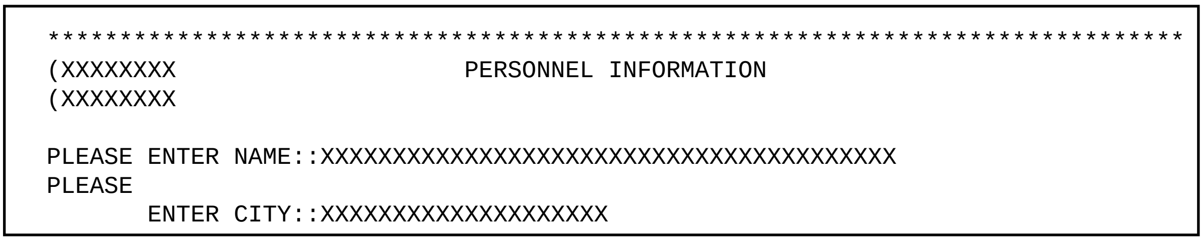
```
*******************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

       ENTER CITY::XXXXXXXXXXXXXXXXXXXX
PLEASE ENTER NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE
```

Enter the line command "..M" as shown below and move the cursor to the position indicated:



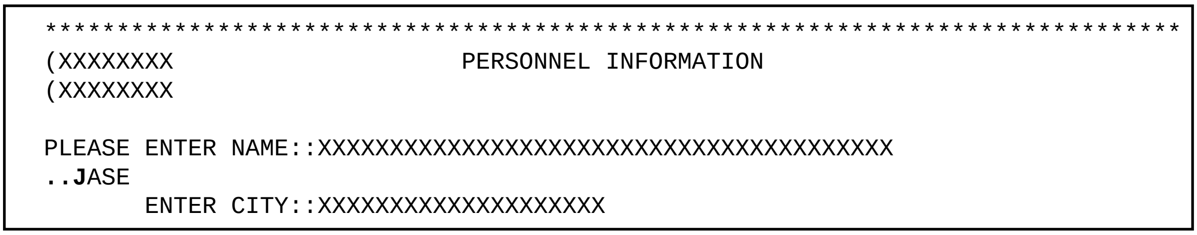
```
*******************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

..M    ENTER CITY::XXXXXXXXXXXXXXXXXXXX
PLEASE ENTER NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
[]LEASE
```

As a result, the line where the command was entered will be moved to the line after the one in which the cursor was positioned:



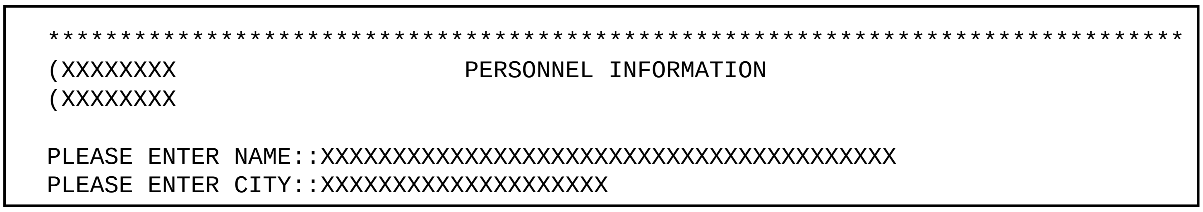
```
*******************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE ENTER NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE
       ENTER CITY::XXXXXXXXXXXXXXXXXXXX
```

Enter the line command "..J" as shown below:



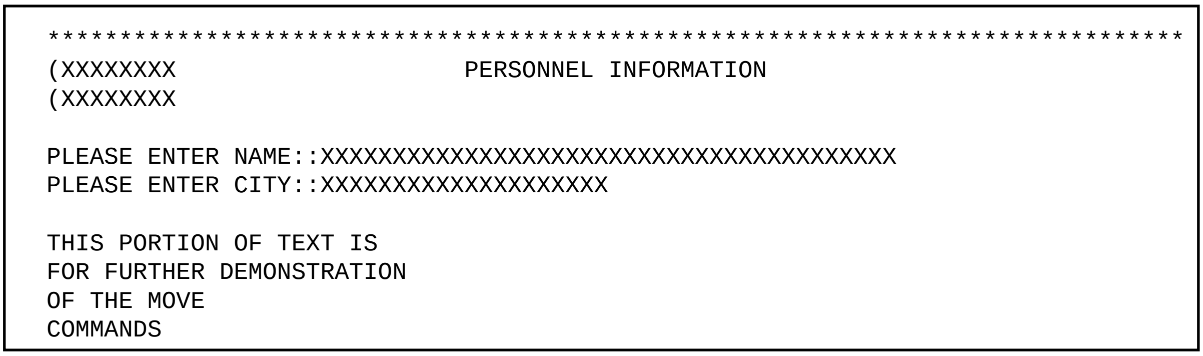
```
*******************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE ENTER NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
..JASE
       ENTER CITY::XXXXXXXXXXXXXXXXXXXX
```

As a result, the line where the command was entered and the line below it will be joined:



```
*******************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE ENTER NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE ENTER CITY::XXXXXXXXXXXXXXXXXXXX
```

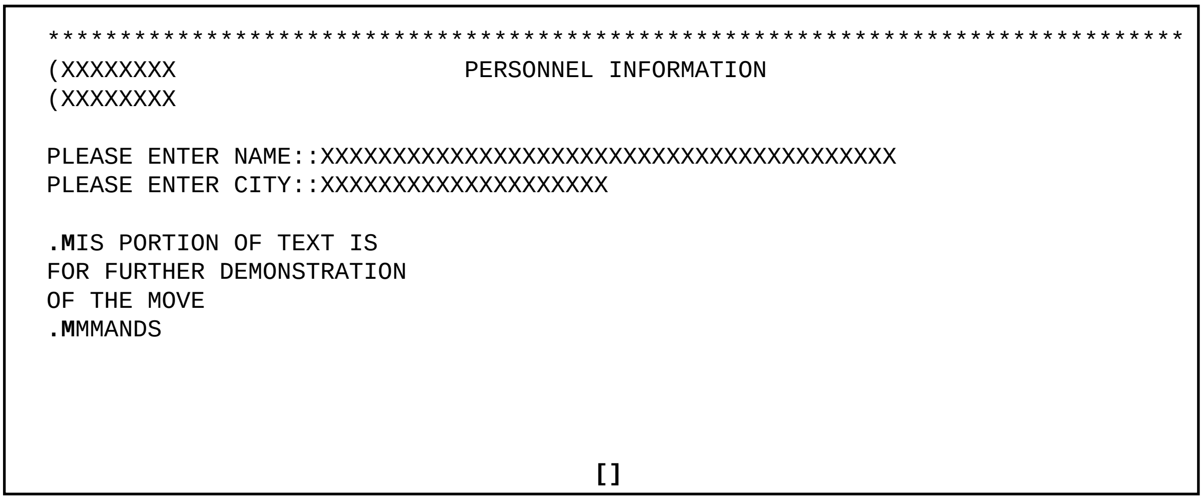
Type in some additional text in the same order and position as below:



```
*******************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE ENTER NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE ENTER CITY::XXXXXXXXXXXXXXXXXXXX

THIS PORTION OF TEXT IS
FOR FURTHER DEMONSTRATION
OF THE MOVE
COMMANDS
```

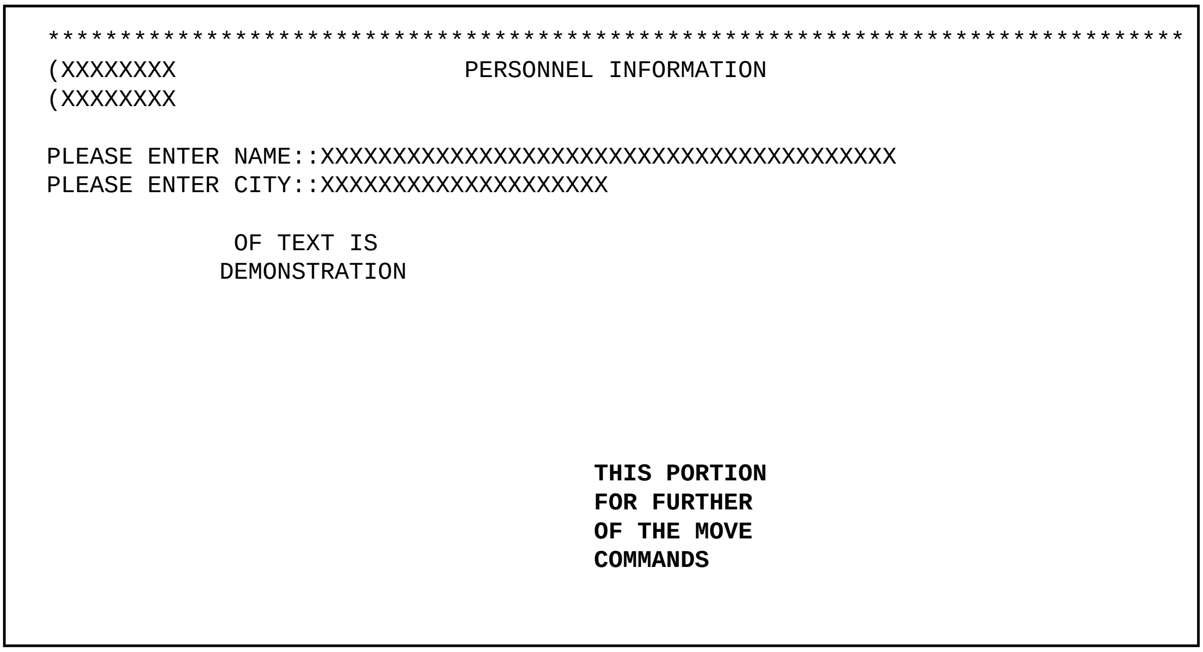
Press ENTER.

Now type in the field command ".M" twice as shown below to move a block of fields:



```
*******************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE ENTER NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE ENTER CITY::XXXXXXXXXXXXXXXXXXXX

.MIS PORTION OF TEXT IS
FOR FURTHER DEMONSTRATION
OF THE MOVE
.MMMANDS




                                      []
```

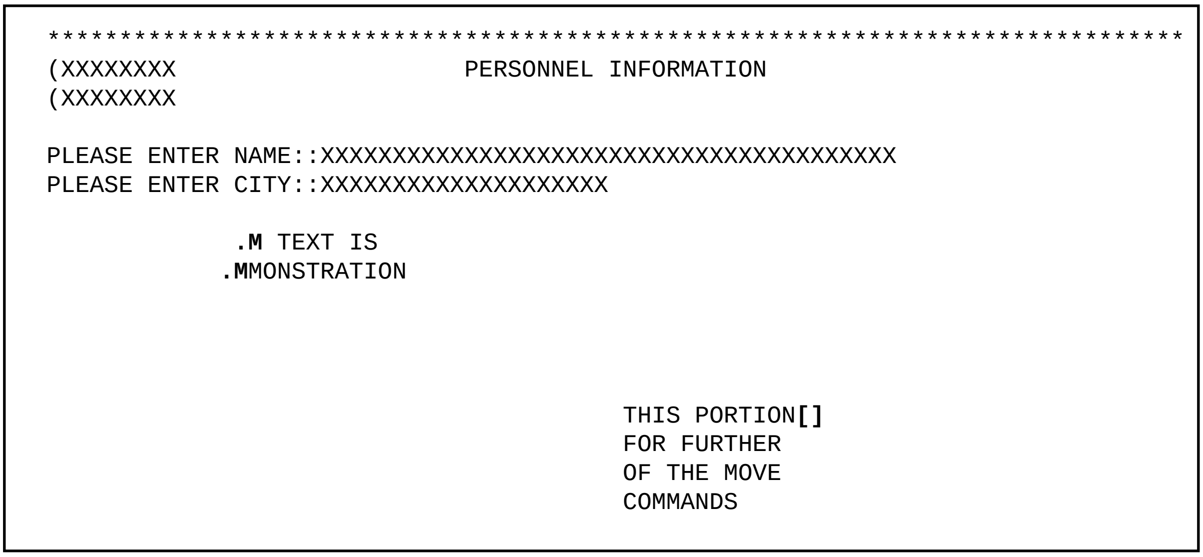
Move the cursor to the position indicated above. As a result, the following block of fields will be moved to the following position, the top left corner of the block being placed at the cursor position:



```
*******************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE ENTER NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE ENTER CITY::XXXXXXXXXXXXXXXXXXXX

             OF TEXT IS
            DEMONSTRATION







                                      THIS PORTION
                                      FOR FURTHER
                                      OF THE MOVE
                                      COMMANDS
```

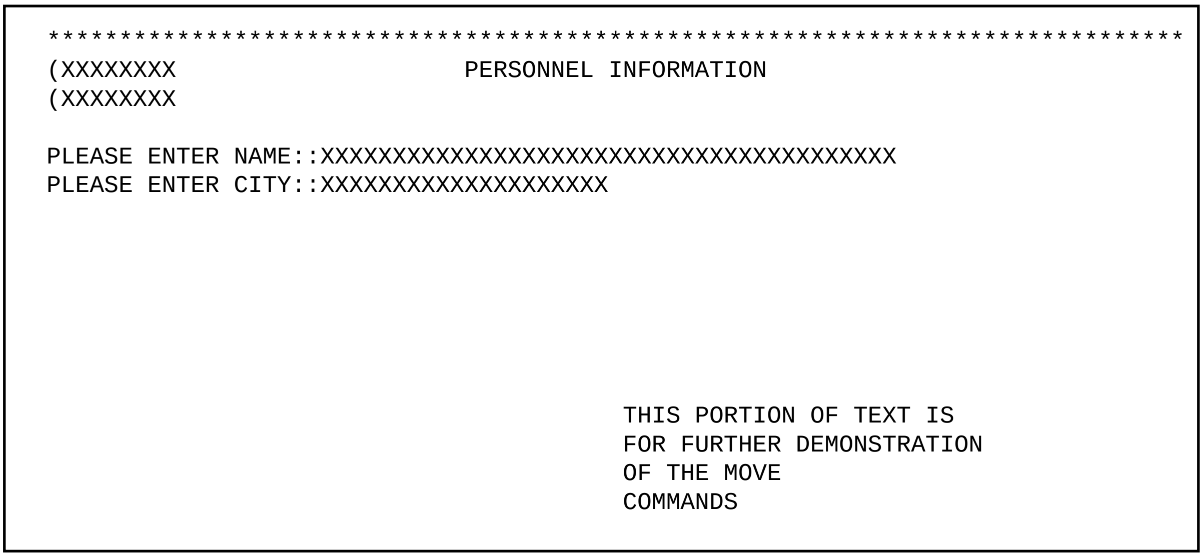
The position and size of the fields where the commands are entered determine the size of the block of fields that is moved, as shown above.

Enter the field command ".M" twice as shown below and move the cursor to the position indicated:



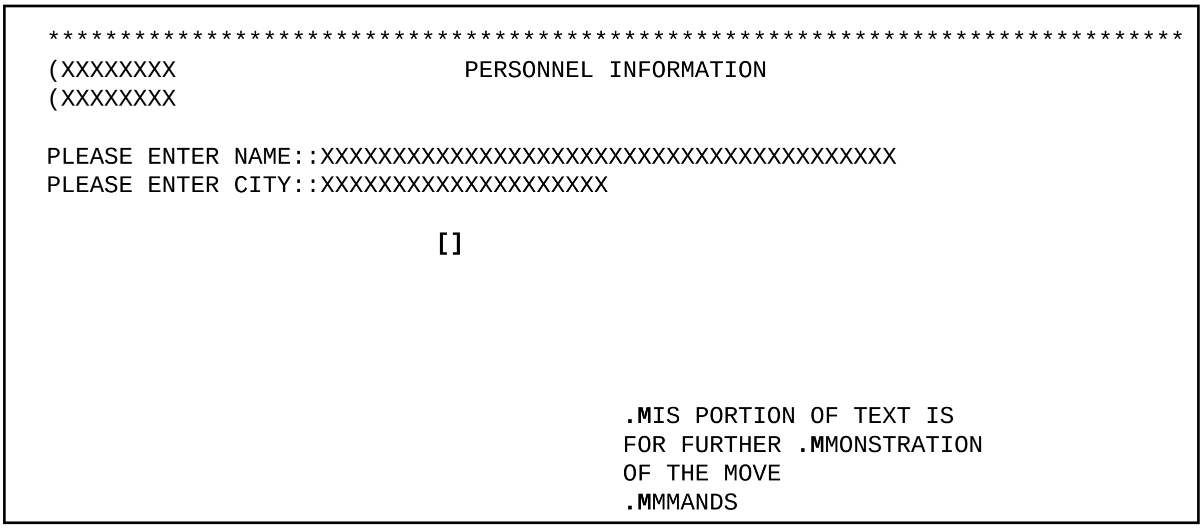
```
*******************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE ENTER NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE ENTER CITY::XXXXXXXXXXXXXXXXXXXX

             .M TEXT IS
            .MMONSTRATION




                                        THIS PORTION[]
                                        FOR FURTHER
                                        OF THE MOVE
                                        COMMANDS
```

The field and block sizes are marked again. Note that the cursor marks the target position of the top left corner of the whole block, **not** that of the top left field within the block. The result will be the following:



```
*******************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE ENTER NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE ENTER CITY::XXXXXXXXXXXXXXXXXXXX







                                        THIS PORTION OF TEXT IS
                                        FOR FURTHER DEMONSTRATION
                                        OF THE MOVE
                                        COMMANDS
```

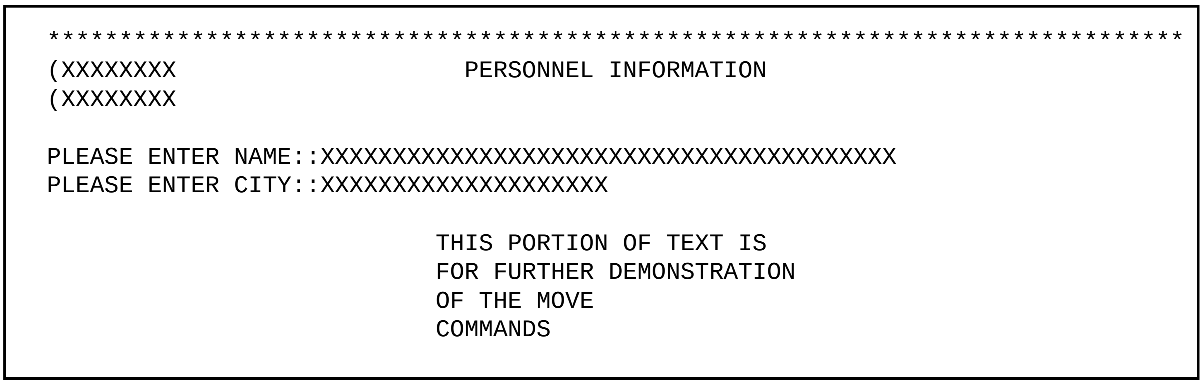
Enter the command ".M" three times to determine the entire block of fields as shown below:



```
*******************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE ENTER NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE ENTER CITY::XXXXXXXXXXXXXXXXXXXX

                           []





                                        .MIS PORTION OF TEXT IS
                                        FOR FURTHER .MMONSTRATION
                                        OF THE MOVE
                                        .MMMANDS
```

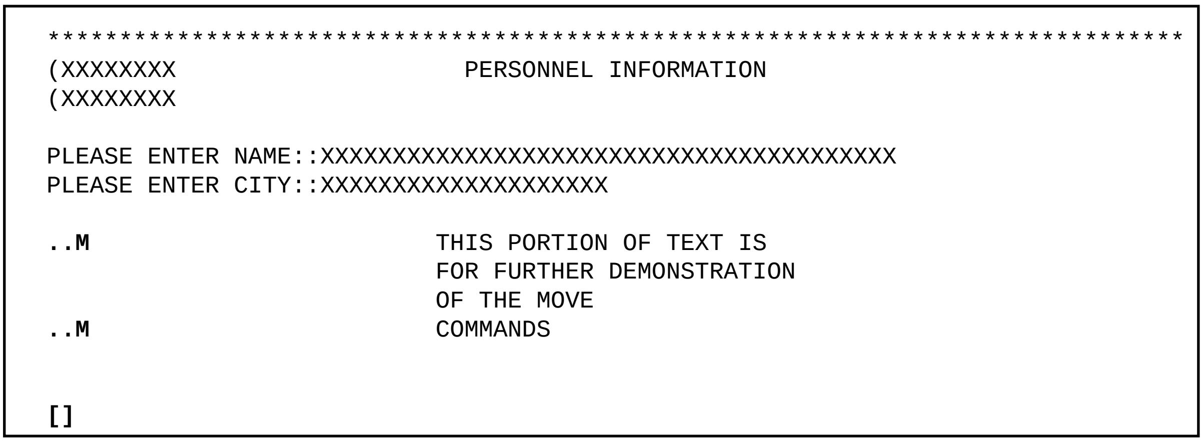
Move the cursor to the position indicated above.

The block of fields will be moved to the position shown below:



```
********************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE ENTER NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE ENTER CITY::XXXXXXXXXXXXXXXXXXXX

                           THIS PORTION OF TEXT IS
                           FOR FURTHER DEMONSTRATION
                           OF THE MOVE
                           COMMANDS
```

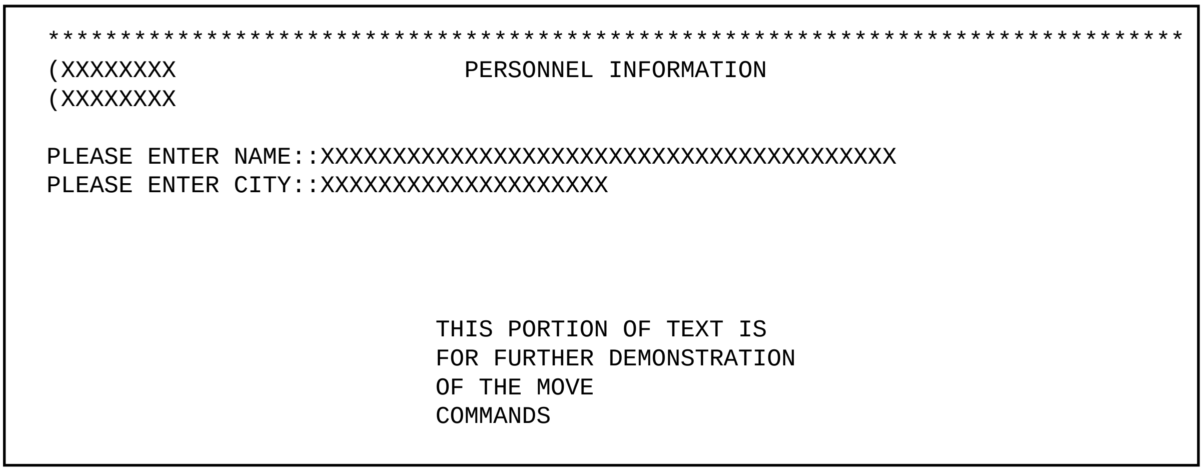
Enter the line command "..M" twice as shown below:



```
********************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE ENTER NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE ENTER CITY::XXXXXXXXXXXXXXXXXXXX

..M                        THIS PORTION OF TEXT IS
                           FOR FURTHER DEMONSTRATION
                           OF THE MOVE
..M                        COMMANDS


[]
```

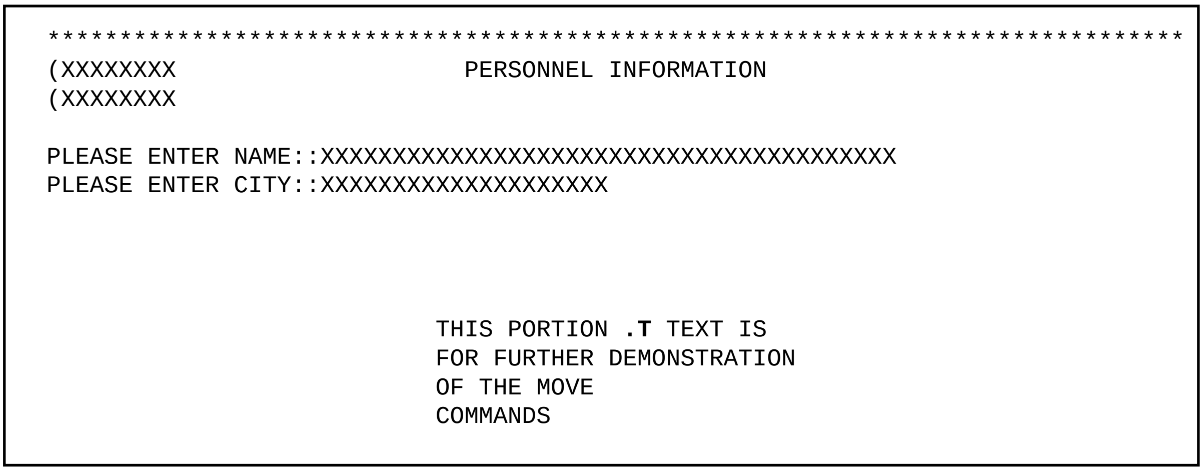
Move the cursor to the position indicated above.

The block of lines marked above will be placed below the line in which the cursor is positioned:



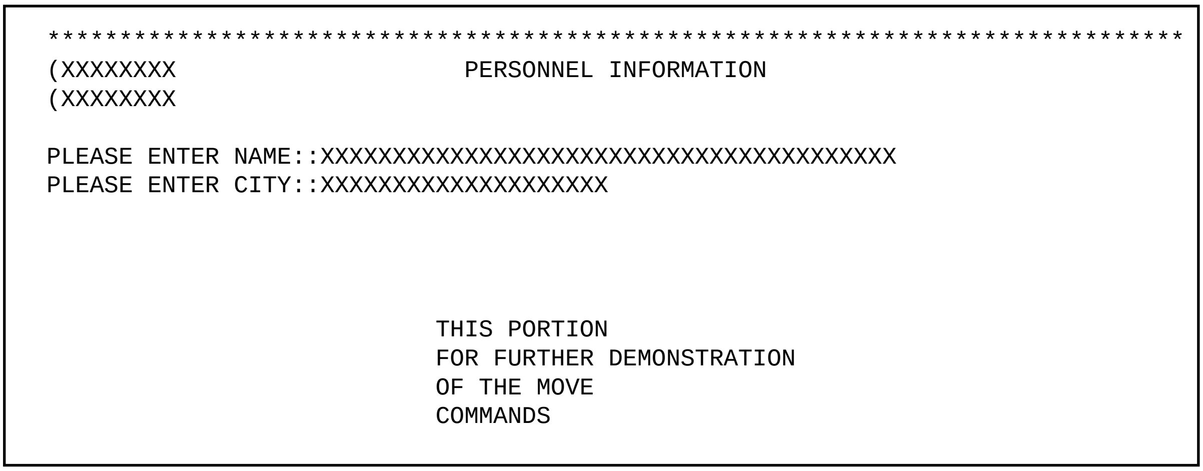
```
********************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE ENTER NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE ENTER CITY::XXXXXXXXXXXXXXXXXXXX




                           THIS PORTION OF TEXT IS
                           FOR FURTHER DEMONSTRATION
                           OF THE MOVE
                           COMMANDS
```

Enter the command ".T" as shown below:



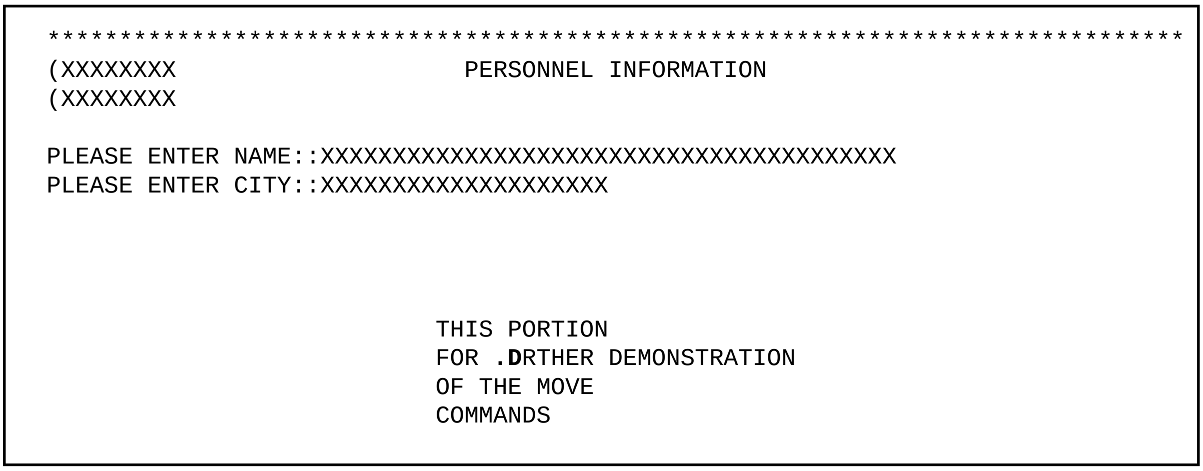
```
*******************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE ENTER NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE ENTER CITY::XXXXXXXXXXXXXXXXXXXX




                           THIS PORTION .T TEXT IS
                           FOR FURTHER DEMONSTRATION
                           OF THE MOVE
                           COMMANDS
```

As a result, the rest of the line, starting from the field in which the command was entered, will be deleted:



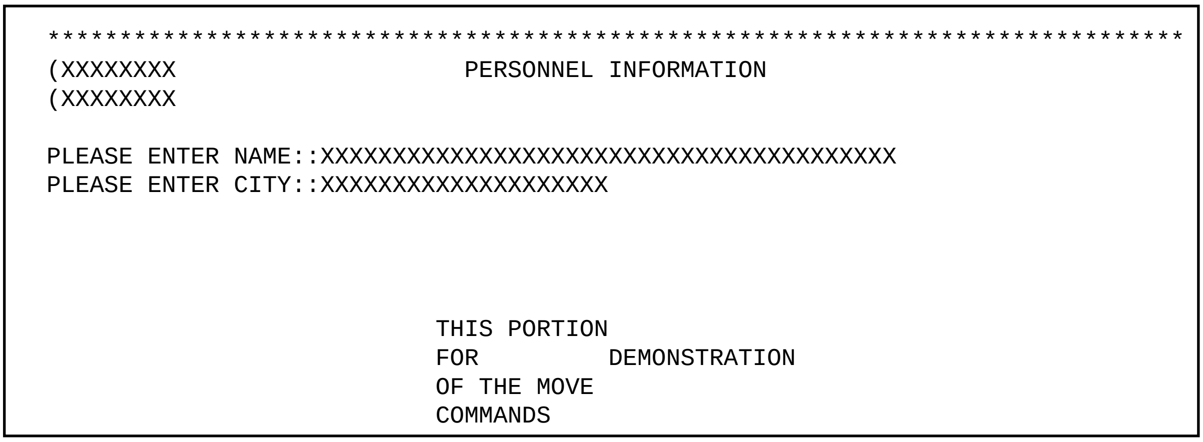
```
*******************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE ENTER NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE ENTER CITY::XXXXXXXXXXXXXXXXXXXX




                           THIS PORTION
                           FOR FURTHER DEMONSTRATION
                           OF THE MOVE
                           COMMANDS
```

Enter the field command ".D" as shown below:



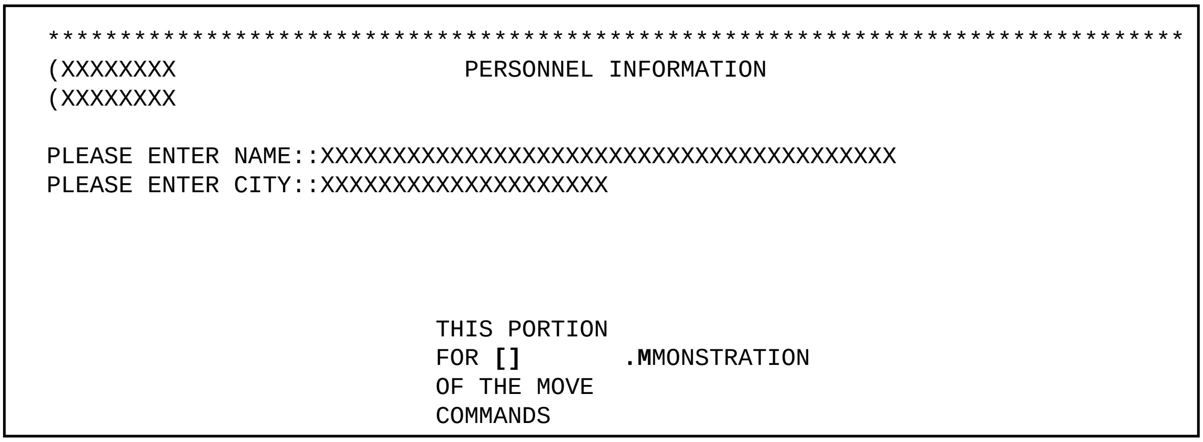
```
*******************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE ENTER NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE ENTER CITY::XXXXXXXXXXXXXXXXXXXX




                           THIS PORTION
                           FOR .DRTHER DEMONSTRATION
                           OF THE MOVE
                           COMMANDS
```

The field marked with the command will be deleted:



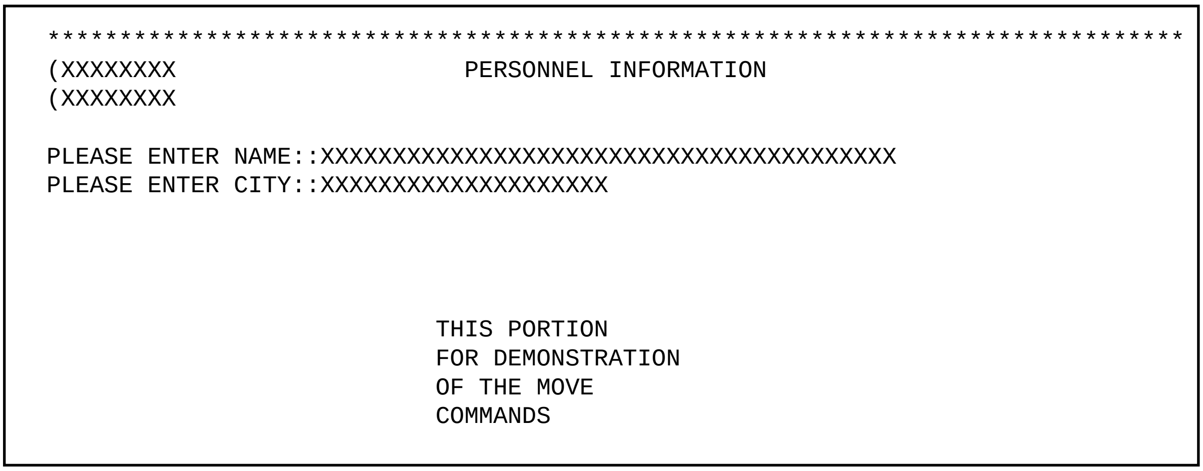
```
********************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE ENTER NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE ENTER CITY::XXXXXXXXXXXXXXXXXXXX




                           THIS PORTION
                           FOR         DEMONSTRATION
                           OF THE MOVE
                           COMMANDS
```

Enter the field command ".M" as shown below; then move the cursor to the position indicated:



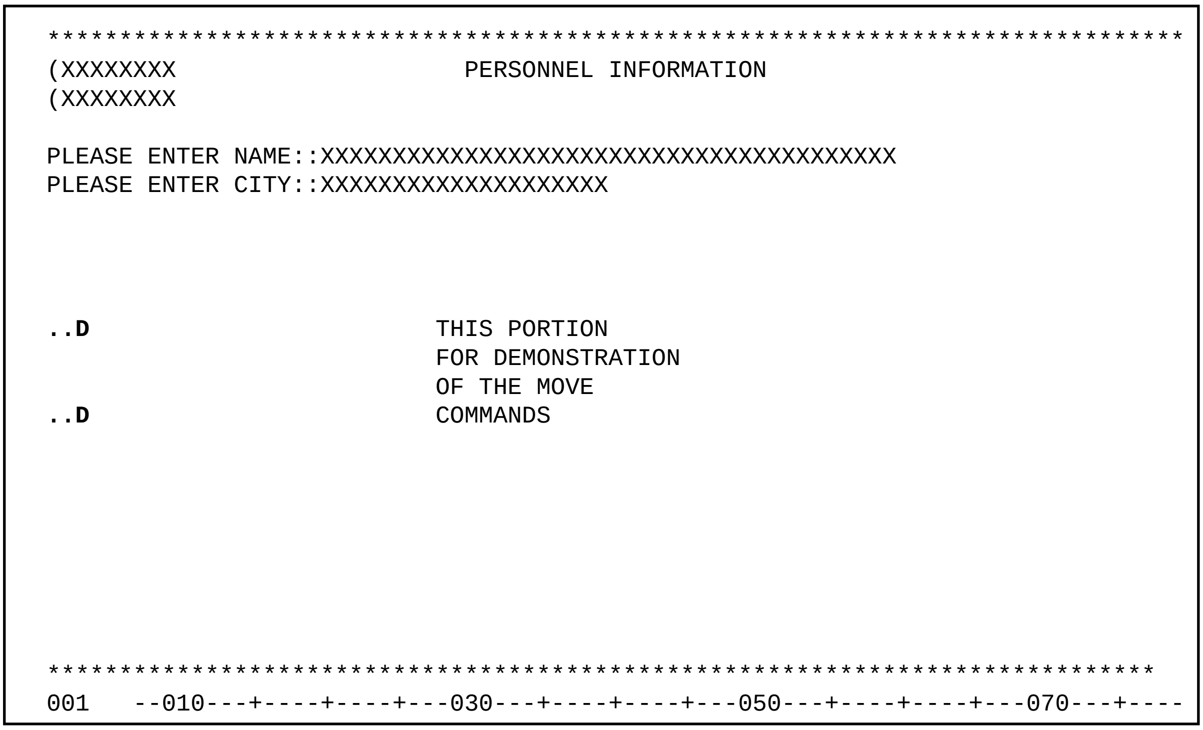
```
********************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE ENTER NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE ENTER CITY::XXXXXXXXXXXXXXXXXXXX




                           THIS PORTION
                           FOR []      .MMONSTRATION
                           OF THE MOVE
                           COMMANDS
```

The field marked with the command will be moved to the cursor position:



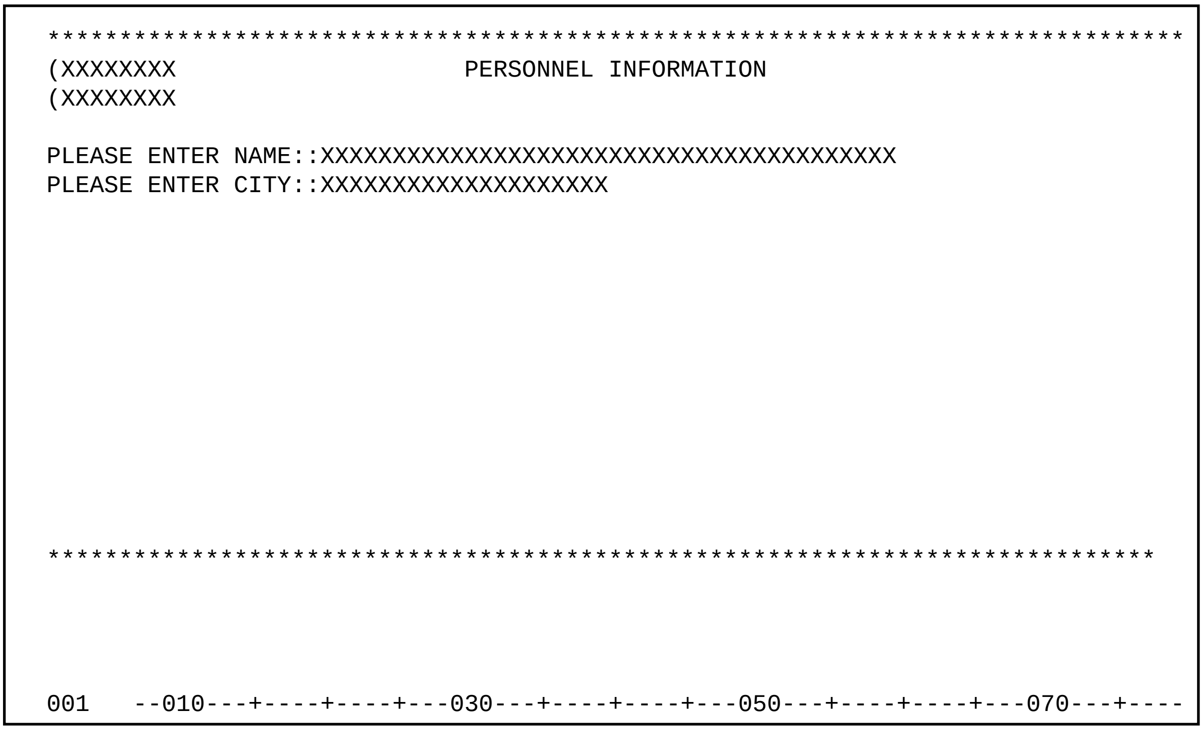
```
********************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE ENTER NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE ENTER CITY::XXXXXXXXXXXXXXXXXXXX




                           THIS PORTION
                           FOR DEMONSTRATION
                           OF THE MOVE
                           COMMANDS
```

Enter the line command "..D" twice as shown below to delete a block of lines:



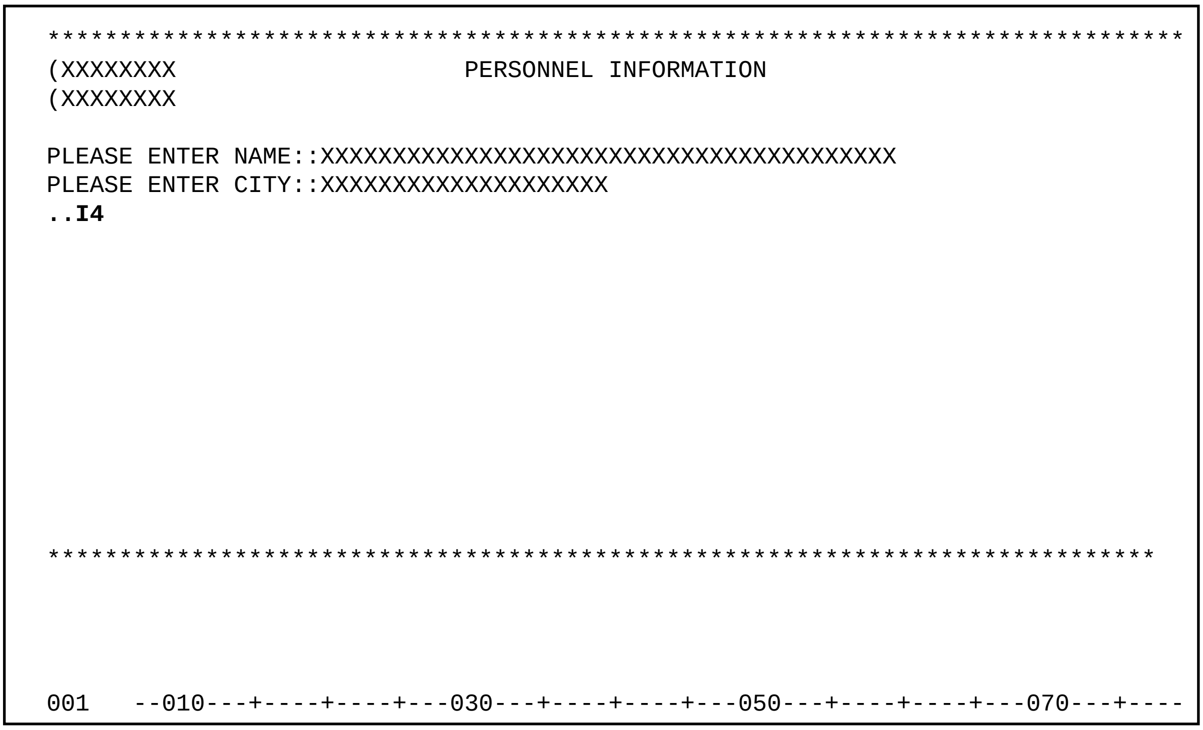
```
********************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE ENTER NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE ENTER CITY::XXXXXXXXXXXXXXXXXXXX




..D                        THIS PORTION
                           FOR DEMONSTRATION
                           OF THE MOVE
..D                        COMMANDS








******************************************************************************
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

The block of lines delimited by the commands will be deleted:



```
********************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE ENTER NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE ENTER CITY::XXXXXXXXXXXXXXXXXXXX












******************************************************************************




001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

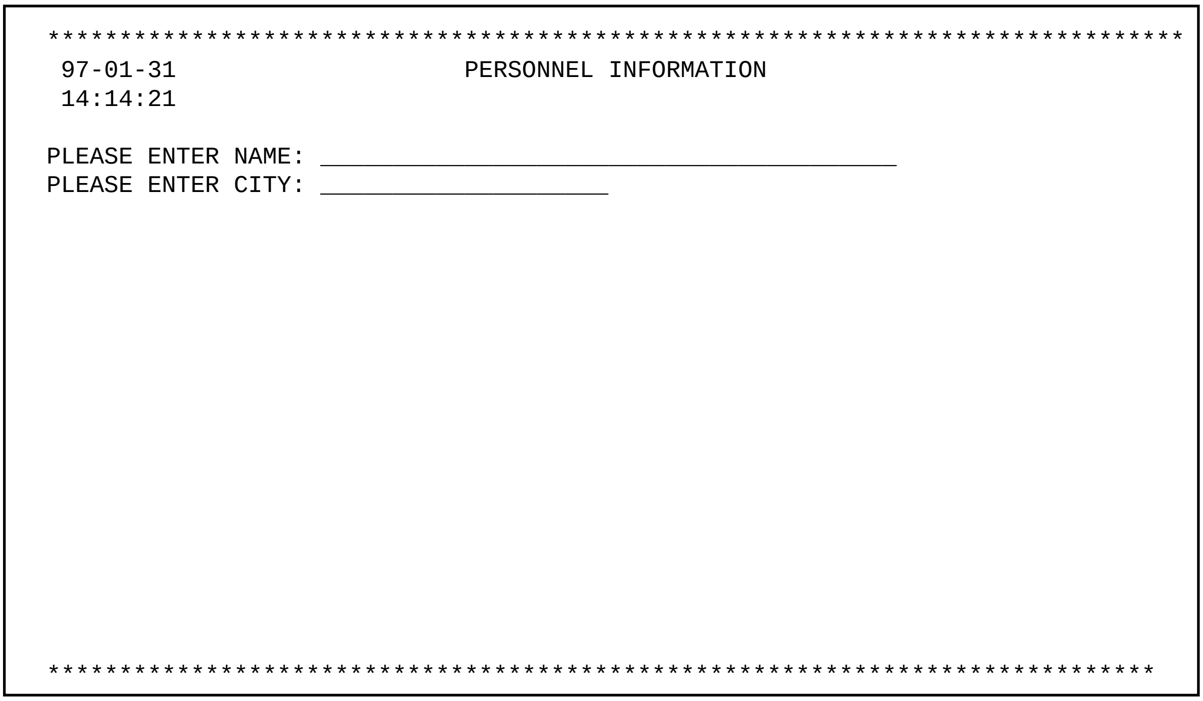
Enter the line command "..I4" as shown below:



```
*******************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE ENTER NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE ENTER CITY::XXXXXXXXXXXXXXXXXXXX
..I4











*****************************************************************************




001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

Four empty lines will be inserted, and the bottom line with the asterisks will be moved four lines down.

Press PF4 to test the map:



```
*******************************************************************************
 97-01-31                    PERSONNEL INFORMATION
 14:14:21

PLEASE ENTER NAME: ________________________________________
PLEASE ENTER CITY: ____________________
















*****************************************************************************
```

Press PF3 to end testing of the map. The map editing screen will appear again.

Press PF3 again to end map editing. The Field and Variable Definitions Summary screen will appear. This will be discussed in a later session.

Press ENTER. The Edit Map menu will appear with Name set to MAP001. Enter "S" in the Code field. The map is now saved in source form.

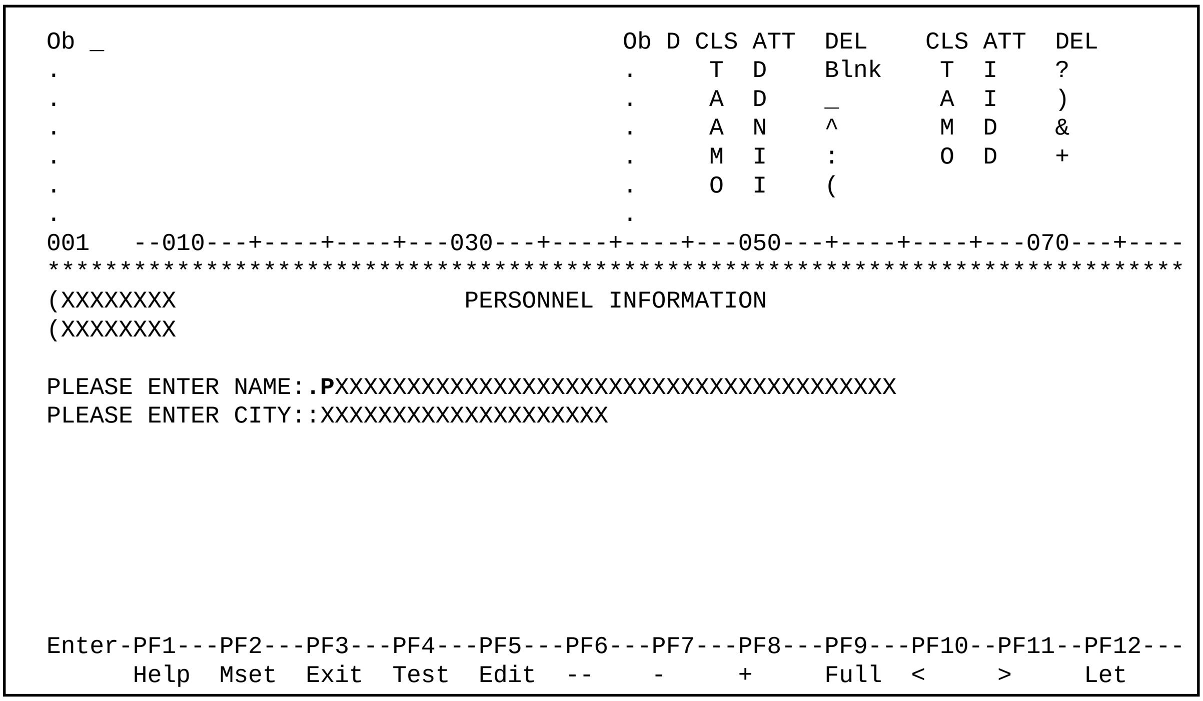
End of Session 1.

Session 2 - Processing Rules

On the Edit Map menu, enter the code "E" and name MAP001 (if it is not already entered).

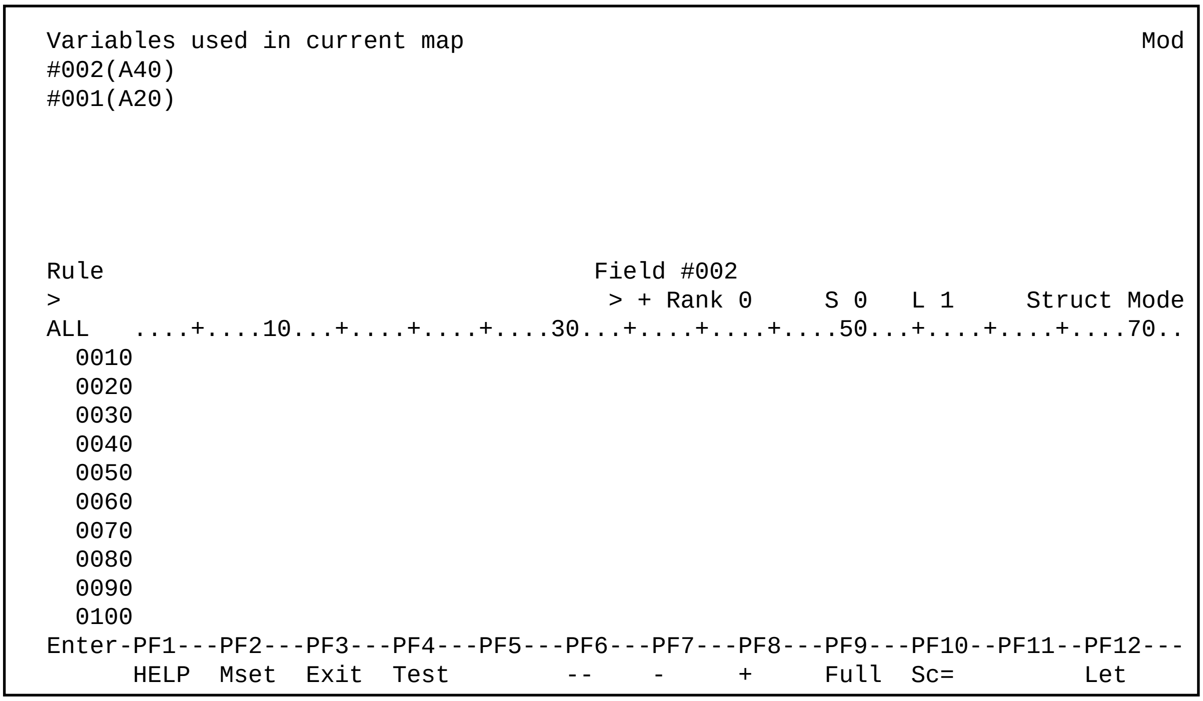
The map editing screen will appear in split-screen mode with map MAP001 being read into the editing area.

Enter the command ".P" as shown below:



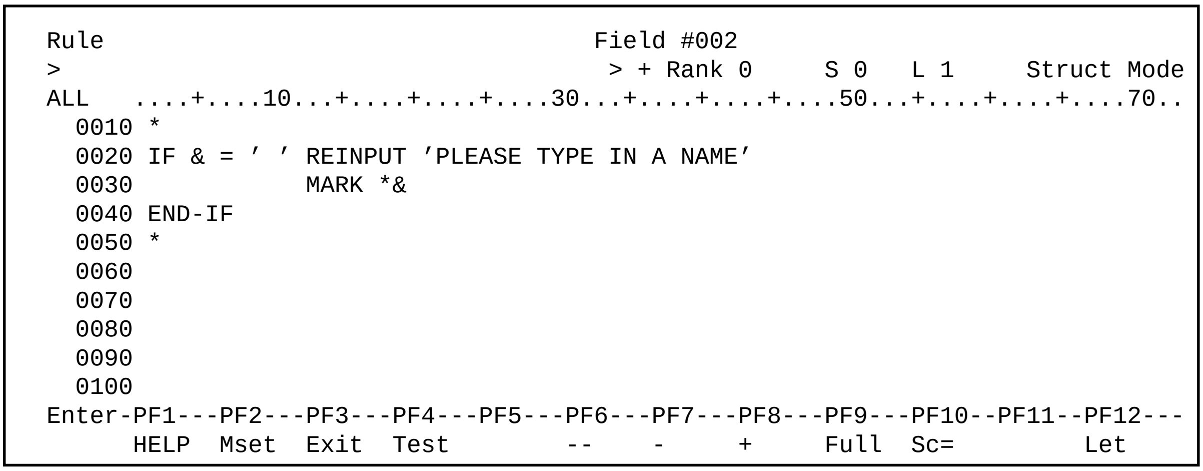
```
Ob _                                    Ob D CLS ATT  DEL    CLS ATT  DEL
.                                       .     T   D    Blnk    T   I    ?
.                                       .     A   D    _       A   I    )
.                                       .     A   N    ^       M   D    &
.                                       .     M   I    :       O   D    +
.                                       .     O   I    (
.                                       .
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
*******************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE ENTER NAME:.PXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE ENTER CITY::XXXXXXXXXXXXXXXXXXXX







Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test  Edit  --    -     +     Full  <     >     Let
```

This will invoke the *processing rule* editor for the field in which the command was entered:



```
Variables used in current map                                               Mod
#002(A40)
#001(A20)





Rule                                 Field #002
>                                     > + Rank 0     S 0   L 1     Struct Mode
ALL   ....+....10...+....+....+....30...+....+....+....50...+....+....+....70..
  0010
  0020
  0030
  0040
  0050
  0060
  0070
  0080
  0090
  0100
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      HELP  Mset  Exit  Test        --    -     +     Full  Sc=         Let
```

Type in the following processing rule:



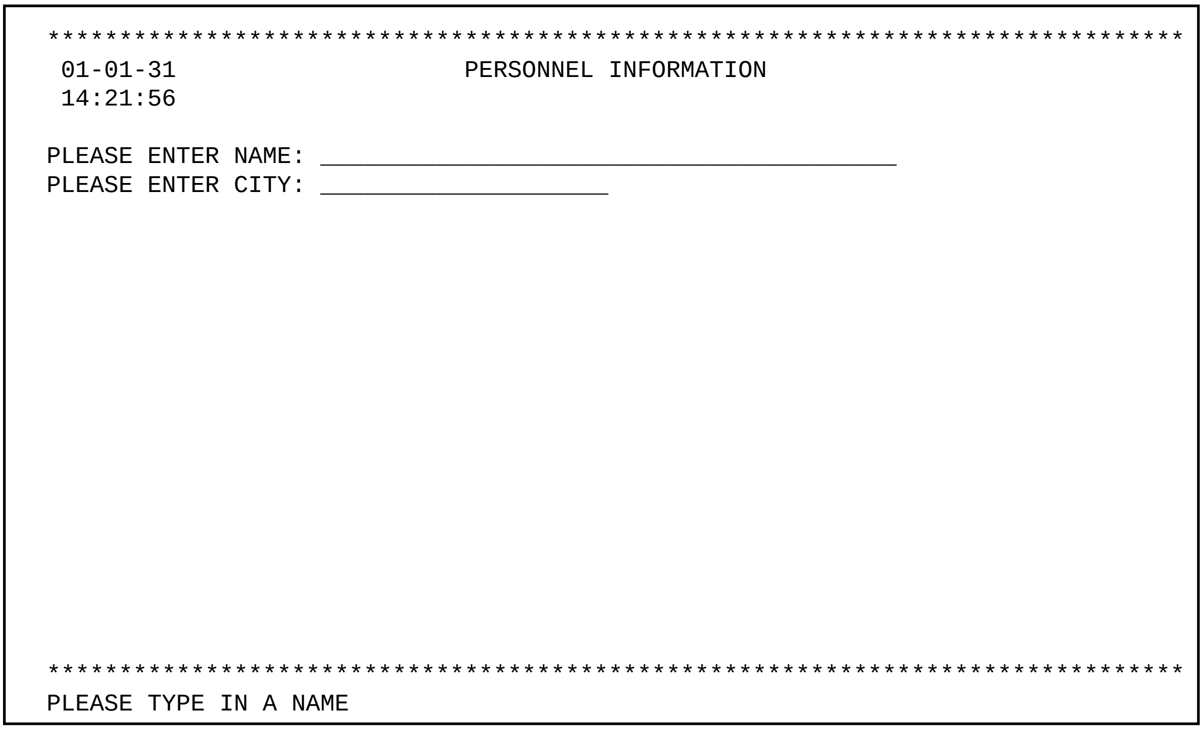
```
Rule                                 Field #002
>                                     > + Rank 0     S 0   L 1     Struct Mode
ALL   ....+....10...+....+....+....30...+....+....+....50...+....+....+....70..
  0010 *
  0020 IF & = ’ ’ REINPUT ’PLEASE TYPE IN A NAME’
  0030            MARK *&
  0040 END-IF
  0050 *
  0060
  0070
  0080
  0090
  0100
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      HELP  Mset  Exit  Test        --    -     +     Full  Sc=         Let
```

The ampersand (&) in the processing rule will be dynamically substituted by the name of the field to which the processing rule is attached.

Press ENTER. Then press PF3 to return to the map editing screen.

Then press PF4 to test the map.

Now you may also test the processing rule: press ENTER. As a result, the processing rule will be executed:



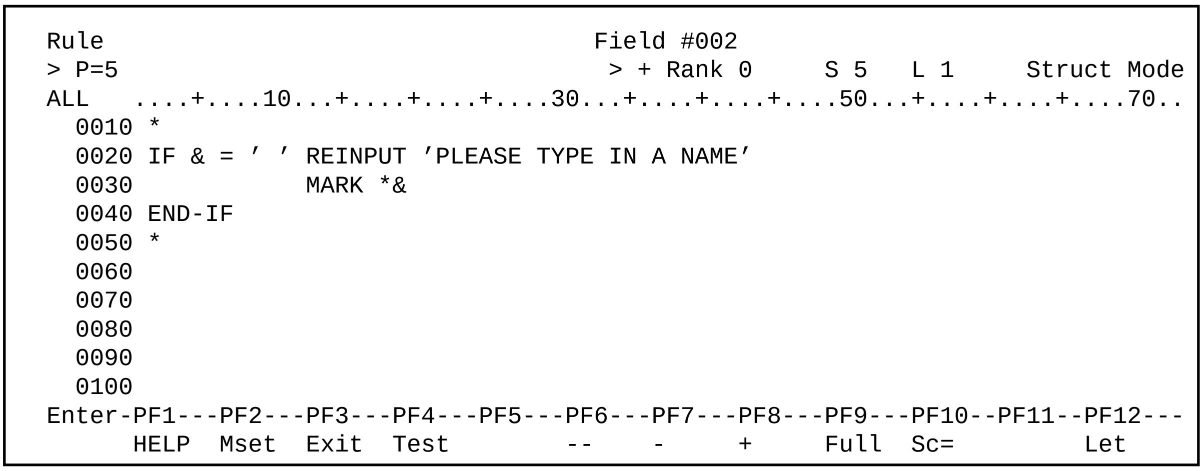
```
********************************************************************************
 01-01-31                    PERSONNEL INFORMATION
 14:21:56

PLEASE ENTER NAME: ________________________________________
PLEASE ENTER CITY: ____________________









********************************************************************************
PLEASE TYPE IN A NAME
```

Note:
The text PLEASE TYPE IN A NAME may not necessarily appear at the bottom of the screen (as shown above) but on another line, depending on the position of the message line as set by the Natural administrator.

Press CLEAR to end testing of the map. The map editing screen will appear again.

Enter the command ".P" in the same position as before. The processing rule for rank 0 of the field where the command was entered will be displayed again.

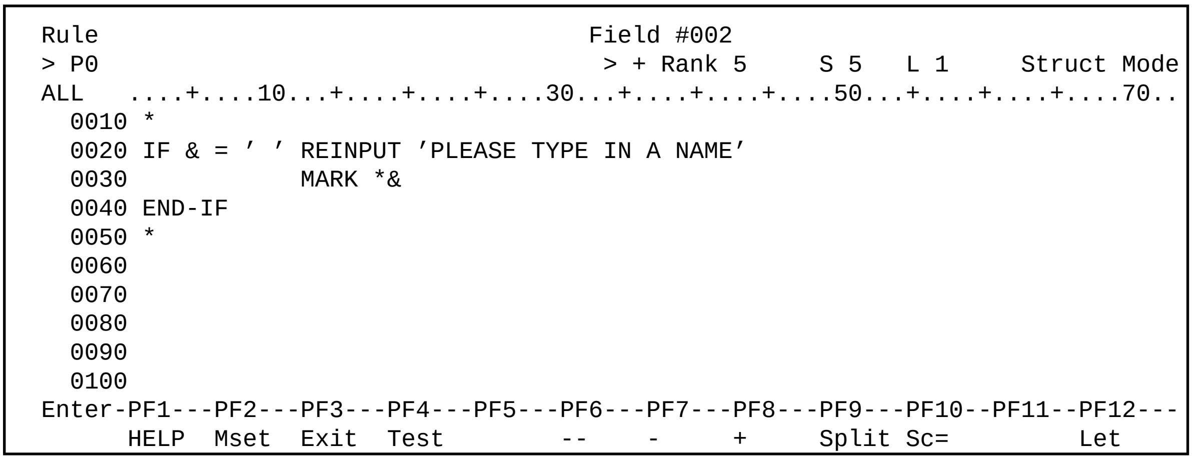
Enter the command "P=5" as shown below:



```
Rule                                  Field #002
> P=5                                  > + Rank 0     S 5   L 1     Struct Mode
ALL   ....+....10...+....+....+....30...+....+....+....50...+....+....+....70..
  0010 *
  0020 IF & = ’ ’ REINPUT ’PLEASE TYPE IN A NAME’
  0030            MARK *&
  0040 END-IF
  0050 *
  0060
  0070
  0080
  0090
  0100
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      HELP  Mset  Exit  Test        --    -     +     Full  Sc=         Let
```

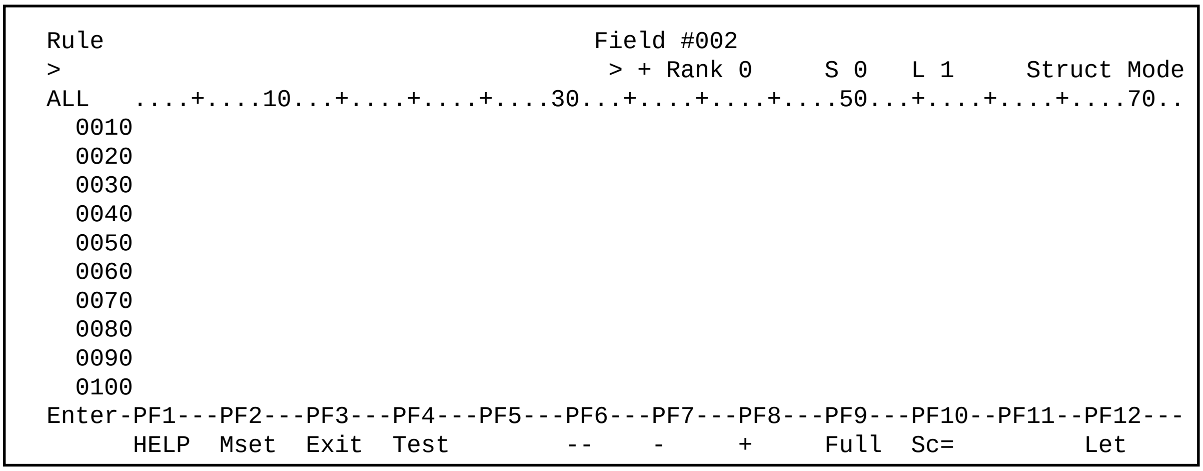
As a result, the processing rule which previously was assigned rank 0 will now be assigned rank 5 (processing rules are processed in ascending order of rank, starting with rank 0).

Enter the command P0 as shown below:



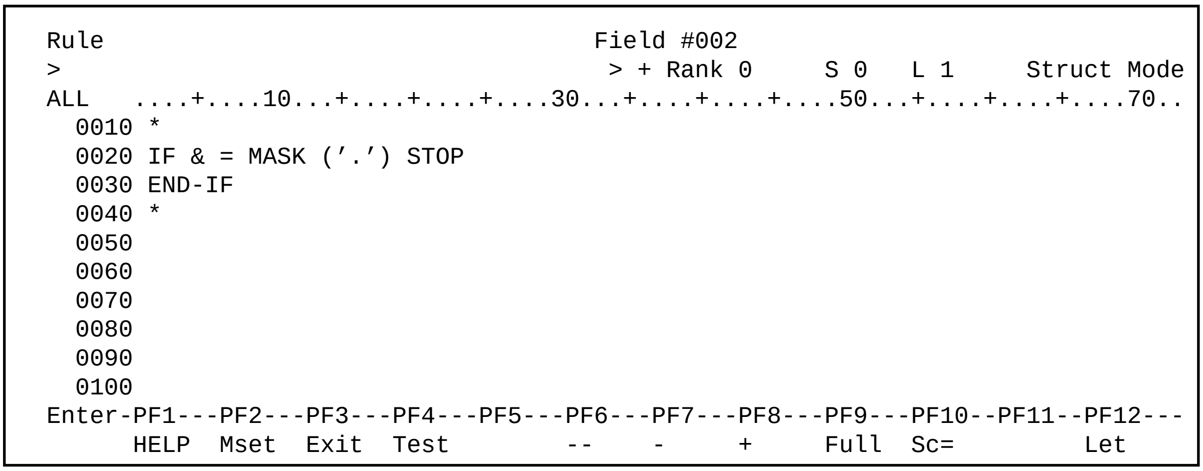
```
Rule                                  Field #002
> P0                                   > + Rank 5     S 5   L 1     Struct Mode
ALL   ....+....10...+....+....+....30...+....+....+....50...+....+....+....70..
  0010 *
  0020 IF & = ’ ’ REINPUT ’PLEASE TYPE IN A NAME’
  0030            MARK *&
  0040 END-IF
  0050 *
  0060
  0070
  0080
  0090
  0100
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      HELP  Mset  Exit  Test        --    -     +     Split Sc=         Let
```

An empty editor screen will be displayed, because there is no longer any processing rule assigned to rank 0.



```
Rule                                  Field #002
>                                      > + Rank 0     S 0   L 1     Struct Mode
ALL   ....+....10...+....+....+....30...+....+....+....50...+....+....+....70..
  0010
  0020
  0030
  0040
  0050
  0060
  0070
  0080
  0090
  0100
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      HELP  Mset  Exit  Test        --    -     +     Full  Sc=         Let
```

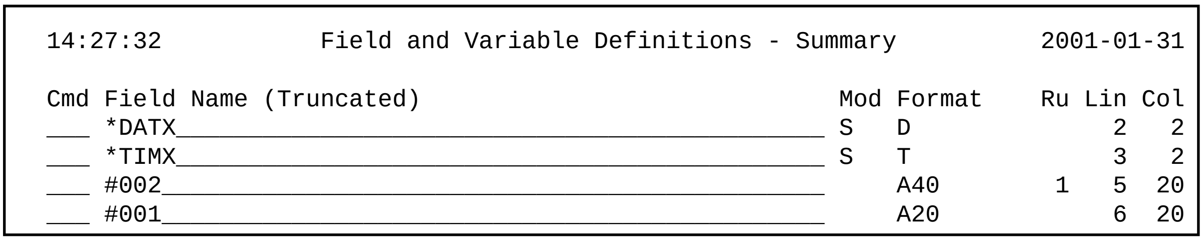
Type in the following processing rule:



```
Rule                                  Field #002
>                                      > + Rank 0     S 0   L 1     Struct Mode
ALL   ....+....10...+....+....+....30...+....+....+....50...+....+....+....70..
  0010 *
  0020 IF & = MASK ('.') STOP
  0030 END-IF
  0040 *
  0050
  0060
  0070
  0080
  0090
  0100
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      HELP  Mset  Exit  Test        --    -     +     Full  Sc=         Let
```

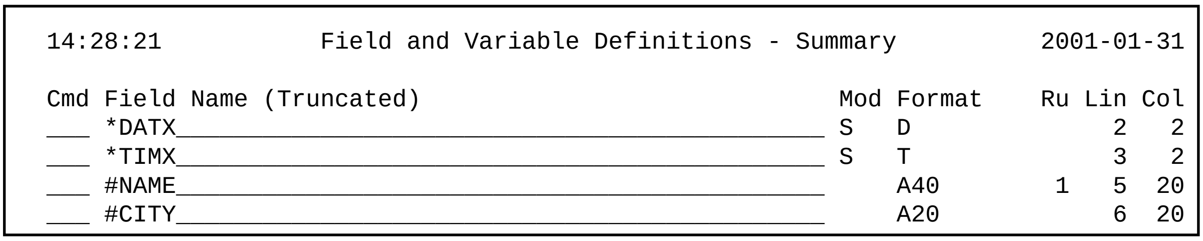
Press ENTER. Then press PF3 to return to the map editing screen.

Press PF3 again. The Field and Variable Definitions Summary screen will be invoked:



```
14:27:32           Field and Variable Definitions - Summary          2001-01-31

Cmd Field Name (Truncated)                             Mod Format     Ru Lin Col
___ *DATX_____________________________________________ S   D               2   2
___ *TIMX_____________________________________________ S   T               3   2
___ #002______________________________________________     A40         1   5  20
___ #001______________________________________________     A20             6  20
```

The fields contained in the map are listed in the order in which they appear on the map. The two user-defined fields are preceded by a hash/number (#). In order to be able to stow the map, you must name these fields. Type in the following names as shown below:



```
14:28:21           Field and Variable Definitions - Summary          2001-01-31

Cmd Field Name (Truncated)                             Mod Format     Ru Lin Col
___ *DATX_____________________________________________ S   D               2   2
___ *TIMX_____________________________________________ S   T               3   2
___ #NAME_____________________________________________     A40         1   5  20
___ #CITY_____________________________________________     A20             6  20
```

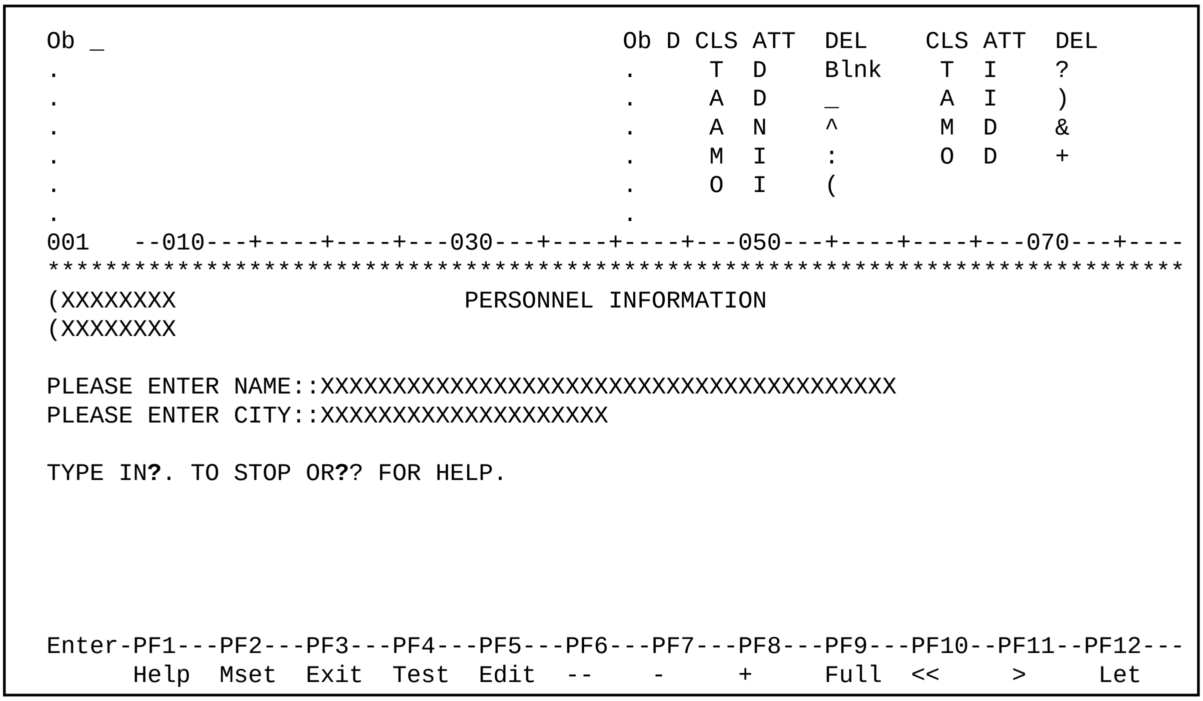
Press ENTER twice. The Edit Map menu will appear. Enter the code "W" to stow the map. The map MAP001 is now stored in source and object form.

End of Session 2.

Session 3 - Extended Field Editing

On the Edit Map menu, enter the code "E" and name MAP001 (if it is not already entered). The map editing screen will appear with map MAP001 being read into the editing area.

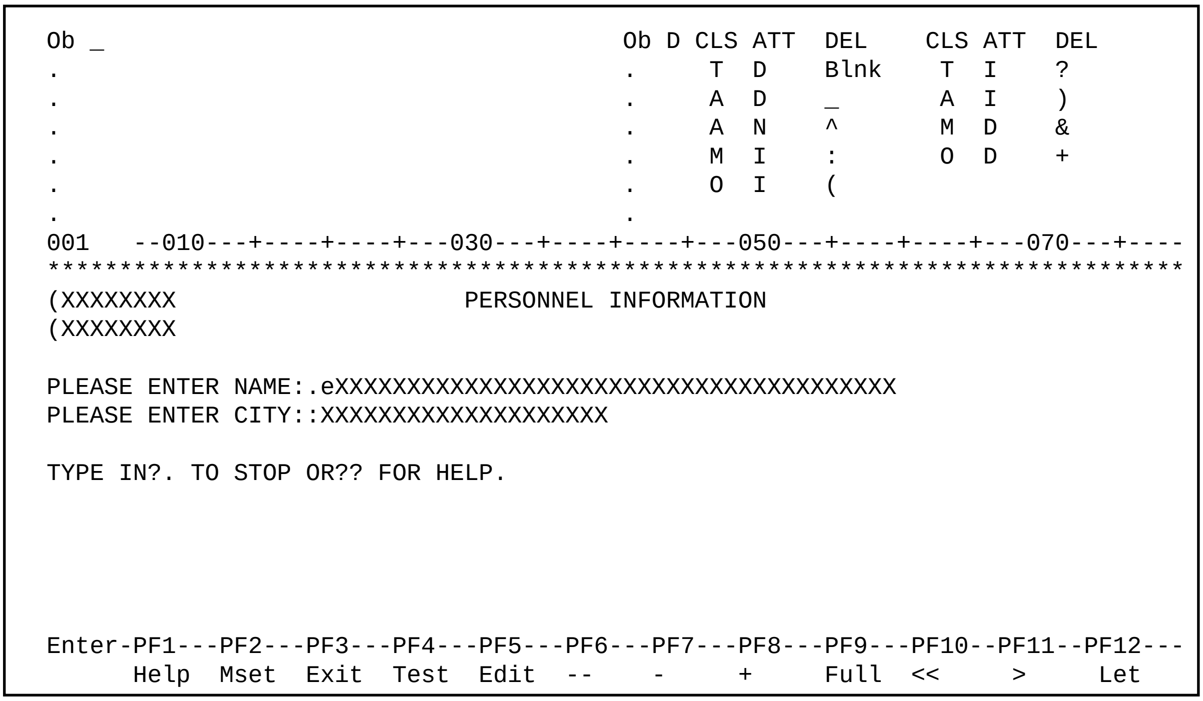
On the map, enter some additional text as shown below:



```
Ob _                                    Ob D CLS ATT  DEL    CLS ATT  DEL
.                                       .     T  D    Blnk    T  I    ?
.                                       .     A  D    _       A  I    )
.                                       .     A  N    ^       M  D    &
.                                       .     M  I    :       O  D    +
.                                       .     O  I    (
.                                       .
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
********************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE ENTER NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE ENTER CITY::XXXXXXXXXXXXXXXXXXXX

TYPE IN?. TO STOP OR?? FOR HELP.





Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test  Edit  --    -     +     Full  <<     >     Let
```

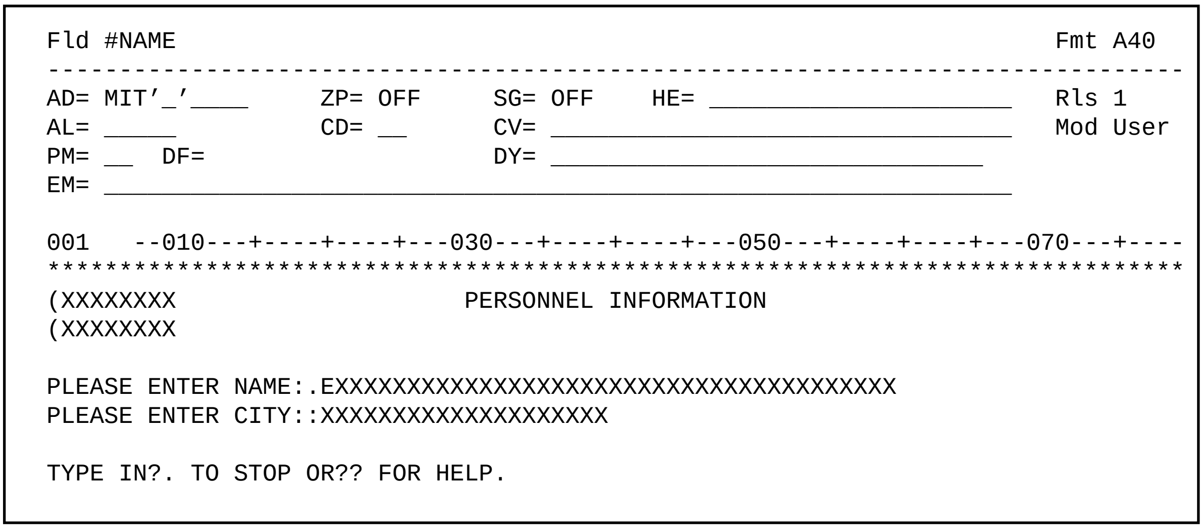
The question mark (?) is the delimiter for intensified text fields.

Then enter the command ".E" as shown below:



```
Ob _                                    Ob D CLS ATT  DEL    CLS ATT  DEL
.                                       .     T  D    Blnk    T  I    ?
.                                       .     A  D    _       A  I    )
.                                       .     A  N    ^       M  D    &
.                                       .     M  I    :       O  D    +
.                                       .     O  I    (
.                                       .
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
*******************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE ENTER NAME:.eXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE ENTER CITY::XXXXXXXXXXXXXXXXXXXX

TYPE IN?. TO STOP OR?? FOR HELP.




Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test  Edit  --    -     +     Full  <<     >     Let
```

The *extended field editing* facility for the field in which the command was entered will appear:



```
Fld #NAME                                                             Fmt A40
-------------------------------------------------------------------------------
AD= MIT'_'____     ZP= OFF     SG= OFF    HE= _____________________   Rls 1
AL= _____          CD= __      CV= ________________________________   Mod User
PM= __  DF=                    DY= ______________________________
EM= _______________________________________________________________

001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
*******************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE ENTER NAME:.EXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE ENTER CITY::XXXXXXXXXXXXXXXXXXXX

TYPE IN?. TO STOP OR?? FOR HELP.
```

In the field "Fmt" enter A20, and in the field "HE=" enter ’HELP001’ (in apostrophes!) as shown below:

```
Fld #NAME                                                               Fmt A20
-------------------------------------------------------------------------------
AD= MIT’_’____     ZP= OFF     SG= OFF    HE= ’HELP001’____________   Rls 1
AL= _____          CD= __      CV= ________________________________   Mod User
PM= __  DF=                    DY= ______________________________
EM= _______________________________________________________________

001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
*******************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE ENTER NAME:.EXXXXXXXXXXXXXXXXXXX
PLEASE ENTER CITY::XXXXXXXXXXXXXXXXXXXX

TYPE IN?. TO STOP OR?? FOR HELP.
```

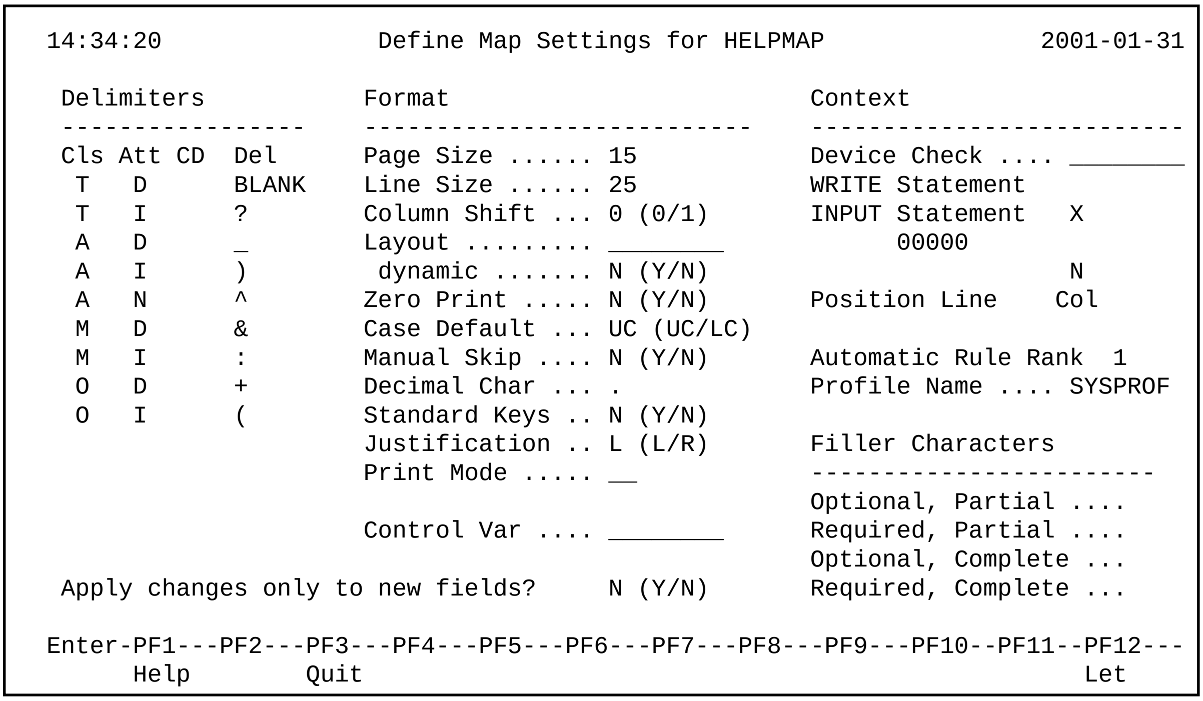
The field length is now reduced to 20. HELP001 (which is yet to be created) is now assigned as helproutine/help map to the field.

Press PF3 to return to the map editing screen. Then press PF3 again to return to the Edit Map menu. Enter the code "W" to stow map MAP001.

On the Edit Map menu, enter the code "H" and name HELP001. The Define Map Settings for HELPMAP screen will be invoked.

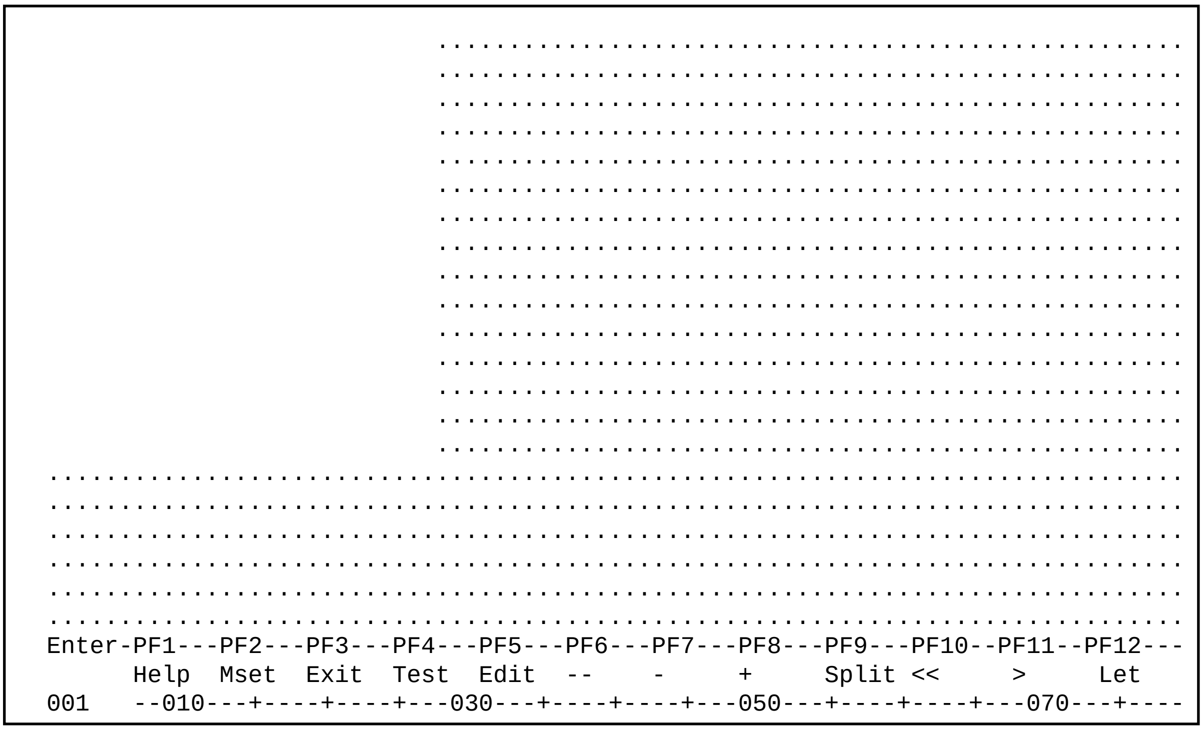
The Page Size is set to 23, the Line Size to 79. Change the Page Size to 15 and the Line Size to 25 by typing over the existing values.

The map settings should now look as below:



```
14:34:20               Define Map Settings for HELPMAP               2001-01-31

 Delimiters           Format                         Context
 -----------------    ---------------------------    --------------------------
 Cls Att CD  Del      Page Size ...... 15            Device Check .... ________
  T   D      BLANK    Line Size ...... 25            WRITE Statement
  T   I      ?        Column Shift ... 0 (0/1)       INPUT Statement   X
  A   D      _        Layout ......... ________            00000
  A   I      )         dynamic ....... N (Y/N)                         N
  A   N      ^        Zero Print ..... N (Y/N)       Position Line    Col
  M   D      &        Case Default ... UC (UC/LC)
  M   I      :        Manual Skip .... N (Y/N)       Automatic Rule Rank  1
  O   D      +        Decimal Char ... .             Profile Name .... SYSPROF
  O   I      (        Standard Keys .. N (Y/N)
                      Justification .. L (L/R)       Filler Characters
                      Print Mode ..... __            ------------------------
                                                     Optional, Partial ....
                      Control Var .... ________      Required, Partial ....
                                                     Optional, Complete ...
 Apply changes only to new fields?     N (Y/N)       Required, Complete ...

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help        Quit                                                  Let
```

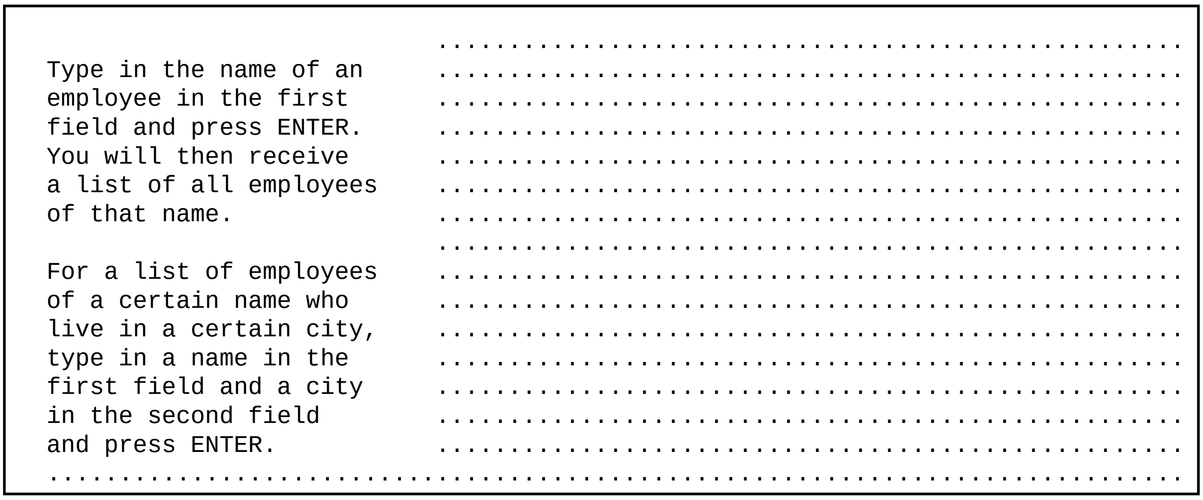
When you press ENTER twice, the map editing screen will appear. Press PF9 to switch to full-screen mode:



```
                           ....................................................
                           ....................................................
                           ....................................................
                           ....................................................
                           ....................................................
                           ....................................................
                           ....................................................
                           ....................................................
                           ....................................................
                           ....................................................
                           ....................................................
                           ....................................................
                           ....................................................
                           ....................................................
                           ....................................................
...............................................................................
...............................................................................
...............................................................................
...............................................................................
...............................................................................
...............................................................................
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test  Edit  --    -     +     Split <<     >     Let
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

The portion of the screen not to be used is filled with lines of periods.

Enter some text as follows:

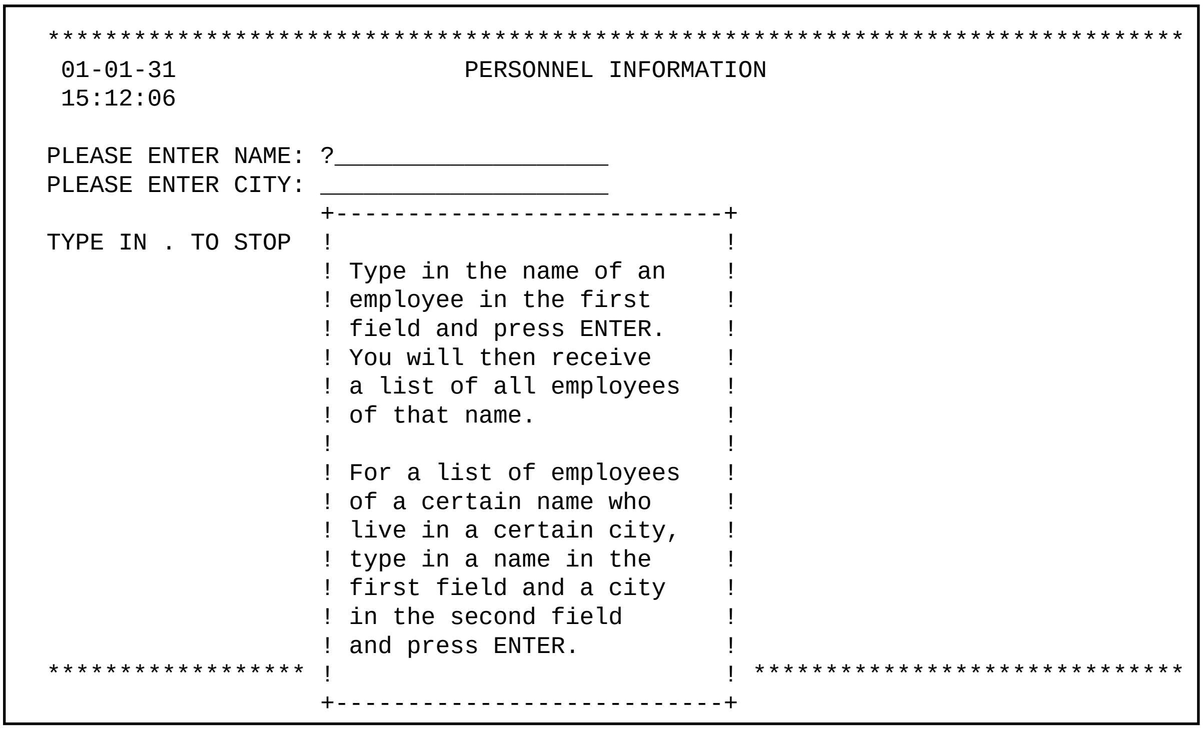


```
                           ....................................................
Type in the name of an     ....................................................
employee in the first      ....................................................
field and press ENTER.     ....................................................
You will then receive      ....................................................
a list of all employees    ....................................................
of that name.              ....................................................
                           ....................................................
For a list of employees    ....................................................
of a certain name who      ....................................................
live in a certain city,    ....................................................
type in a name in the      ....................................................
first field and a city     ....................................................
in the second field        ....................................................
and press ENTER.           ....................................................
...............................................................................
```

Then press PF3 to return to the Edit Map menu. Enter the code "W" to stow help map HELP001.

Enter the code "T" and name MAP001 to test MAP001.

Enter a question mark (?) in the first position of the field entitled PLEASE ENTER NAME. Help map HELP001 will be displayed:



```
********************************************************************************
 01-01-31                    PERSONNEL INFORMATION
 15:12:06

PLEASE ENTER NAME: ?___________________
PLEASE ENTER CITY: ____________________
                   +---------------------------+
TYPE IN . TO STOP  !                           !
                   ! Type in the name of an    !
                   ! employee in the first     !
                   ! field and press ENTER.    !
                   ! You will then receive     !
                   ! a list of all employees   !
                   ! of that name.             !
                   !                           !
                   ! For a list of employees   !
                   ! of a certain name who     !
                   ! live in a certain city,   !
                   ! type in a name in the     !
                   ! first field and a city    !
                   ! in the second field       !
                   ! and press ENTER.          !
****************** !                           ! *******************************
                   +---------------------------+
```

Press ENTER.

Press ENTER again to test the processing rule for the first field.

Press CLEAR to end testing. The Edit Map menu will appear again.

End of Session 3.

Session 4 - INPUT USING MAP

- If you have access to a copy of the program PROG001, enter EDIT PROG001 in the Command line of the Edit Map menu. (By default, sample programs are provided in the system library SYSEXPG; ask your natural administrator for details.)
  PROG001 will be read into the program editor. Make sure that the program is identical to the one shown below.
- If you do not have access to a copy of PROG001, enter EDIT PROGRAM in the Command line of the Edit Map menu.
  The program editor will be invoked. If necessary, CLEAR the program editor (with the command CLEAR). Then type in the following program:

PROG001:

```
** PROG001
**********
DEFINE DATA LOCAL
01 #NAME (A20)
01 #CITY (A20)
01 PERS-VIEW VIEW OF EMPLOYEES
   02 NAME
   02 FIRST-NAME
   02 CITY
END-DEFINE
*
REPEAT
*
INPUT USING MAP ’MAP001’
*
IF #CITY NE ’ ’ AND #NAME NE ’ ’
  FIND PERS-VIEW WITH NAME = #NAME AND CITY = #CITY
    IF NO RECORDS FOUND
      REINPUT ’NO ONE BY THIS NAME LIVING IN THIS CITY.’
        MARK *#CITY
    END-NOREC
    DISPLAY NOTITLE NAME FIRST-NAME CITY
  END-FIND
ELSE
  IF #NAME NE ’ ’
    FIND PERS-VIEW WITH NAME = #NAME
      IF NO RECORDS FOUND
        REINPUT ’PLEASE TRY ANOTHER NAME.’
      END-NOREC
      DISPLAY NOTITLE NAME FIRST-NAME CITY
    END-FIND
  END-IF
END-IF
*
END-REPEAT
END
```

CHECK the program and correct any errors. STOW the program under the name of PROG001 (If no program name is displayed in the top line of the editor, enter STOW PROG001. If the program name PROG001 is displayed, simply enter the command STOW).

Then enter the command RUN to execute the program. Map MAP001 will be displayed.

To see if everything works as intended

1. Press ENTER without typing in anything. As a result, the message PLEASE TYPE IN A NAME will be displayed.
2. Enter a question mark (?) in the first input field of the map. As a result, the help map "Type in the name of ... etc." will appear as a window on the map. - Press ENTER.
3. Enter the name MCKENNA in the first input field of the map. As a result, the message PLEASE TRY ANOTHER NAME will be displayed.
4. Enter the name JONES in the first input field of the map. As a result, the program will display the following list:

   ```
           NAME              FIRST-NAME              CITY
   -------------------- -------------------- --------------------

   JONES                VIRGINIA             TULSA
   JONES                MARSHA               MOBILE
   JONES                ROBERT               MILWAUKEE
   JONES                LILLY                BEVERLEY HILLS
   JONES                EDWARD               CAMDEN
   JONES                MARTHA               KALAMAZOO
   JONES                LAUREL               BALTIMORE
   JONES                KEVIN                DERBY
   JONES                GREGORY              NOTTINGHAM
   ```

   Keep pressing ENTER until you return to the map.

5. Enter the name JONES in the first input field and the name of the city DUNFERMLINE in the second input field. As a result, the message NO ONE BY THIS NAME LIVING IN THIS CITY will be displayed.
6. Enter the name JONES in the first input field and the name of the town TULSA in the second input field. As a result, the program will display the following list:

   ```
           NAME              FIRST-NAME              CITY
   -------------------- -------------------- --------------------

   JONES                VIRGINIA             TULSA
   ```

   Press ENTER to return to the map.

7. Enter a period (.) in the first input field. The program will be terminated, and you will be returned to the program editor.

End of Session 4.

Session 5 - WRITE USING MAP, Fields from a View

Enter the command SAVE PROG002 to save a copy of program PROG001 under the new name of PROG002.

Then enter the command READ PROG002 to read the newly created program PROG002 into the work area.

Modify the program to match with the one on the next page.

PROG002:

```
** PROG002
**********
DEFINE DATA LOCAL
01 #NAME (A20)
01 #CITY (A20)
01 PERS-VIEW VIEW OF EMPLOYEES
   02 NAME
   02 FIRST-NAME
   02 CITY
END-DEFINE
*
REPEAT
*
INPUT USING MAP ’MAP001’
*
IF #CITY NE ’ ’ AND #NAME NE ’ ’
  FIND PERS-VIEW WITH NAME = #NAME AND CITY = #CITY
    IF NO RECORDS FOUND
      REINPUT ’NO-ONE BY THIS NAME LIVING IN THIS CITY.’
        MARK *#CITY
    END-NOREC
*
    AT START OF DATA
      WRITE ’THE FOLLOWING EMPLOYEES LIVE IN’ CITY
    END-START
    WRITE USING MAP ’MAP003’
*
  END-FIND
ELSE
  IF #NAME NE ’ ’
    FIND PERS-VIEW WITH NAME = #NAME
      IF NO RECORDS FOUND
        REINPUT ’PLEASE TRY ANOTHER NAME.’
      END-NOREC
*
      WRITE USING MAP ’MAP002’
*
    END-FIND
  END-IF
END-IF
*
END-REPEAT
END
```

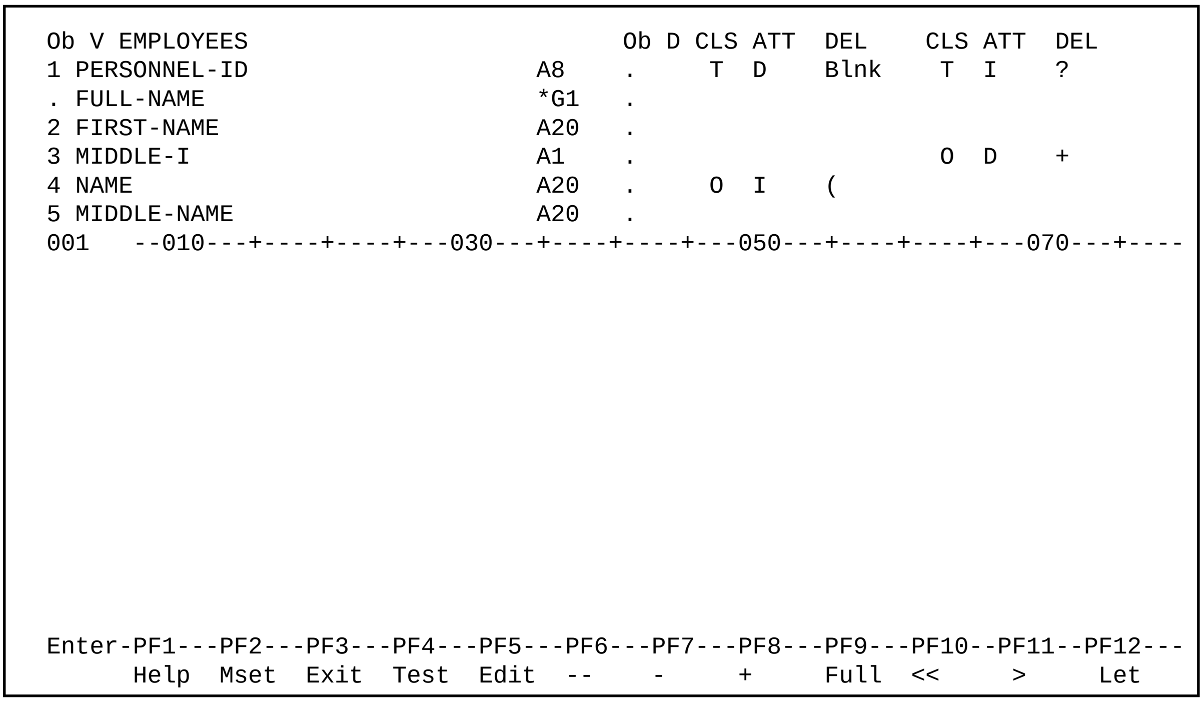
When you have made all changes, enter the command SAVE in the command line of the program editor to save PROG002.

In the command line of the program editor, enter the command EDIT MAP.

The Edit Map menu will be displayed. Enter the code "I" and name MAP002.

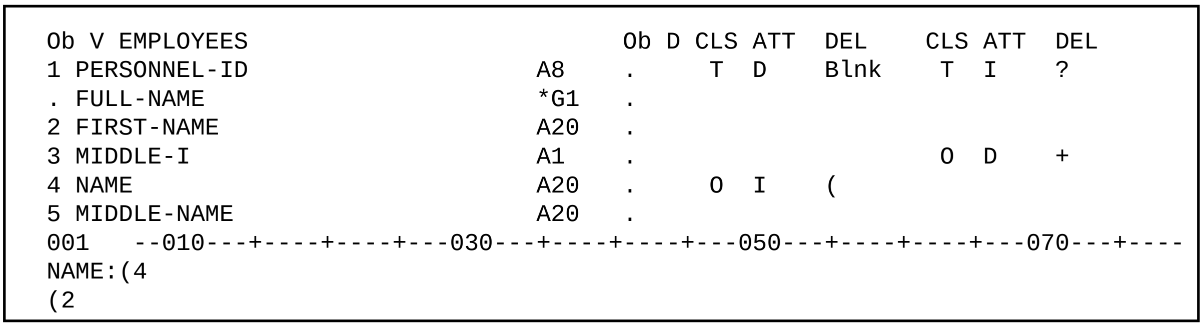
The Define Map Settings For Map screen will be displayed. Change the Page Size to 60. Then type in an "X" after "WRITE Statement" and type a blank over the "X" after "INPUT Statement".

When you press ENTER, the map editing screen will be displayed. In the top line of the screen, enter "V EMPLOYEES". The fields definitions of the view (DDM) EMPLOYEES will be listed:



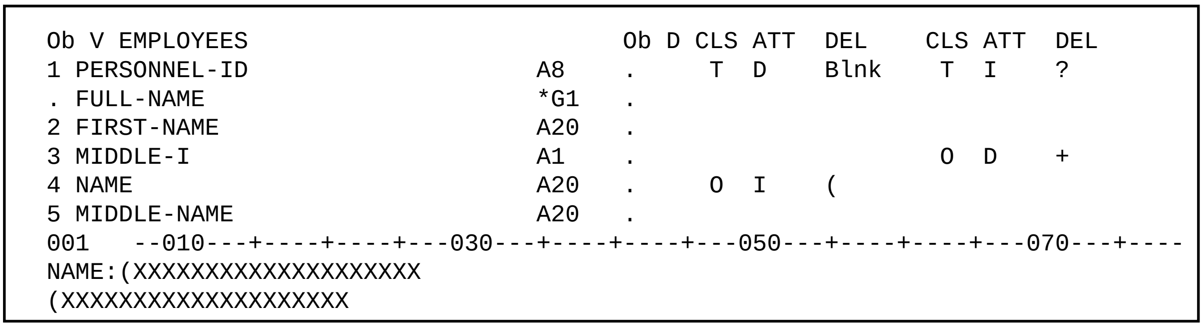
```
Ob V EMPLOYEES                          Ob D CLS ATT  DEL    CLS ATT  DEL
1 PERSONNEL-ID                    A8    .     T  D    Blnk    T  I    ?
. FULL-NAME                       *G1   .
2 FIRST-NAME                      A20   .
3 MIDDLE-I                        A1    .                     O  D    +
4 NAME                            A20   .     O  I    (
5 MIDDLE-NAME                     A20   .
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----












Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test  Edit  --    -     +     Full  <<     >     Let
```

In the editing area, enter the following:



```
Ob V EMPLOYEES                          Ob D CLS ATT  DEL    CLS ATT  DEL
1 PERSONNEL-ID                    A8    .     T  D    Blnk    T  I    ?
. FULL-NAME                       *G1   .
2 FIRST-NAME                      A20   .
3 MIDDLE-I                        A1    .                     O  D    +
4 NAME                            A20   .     O  I    (
5 MIDDLE-NAME                     A20   .
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
NAME:(4
(2
```

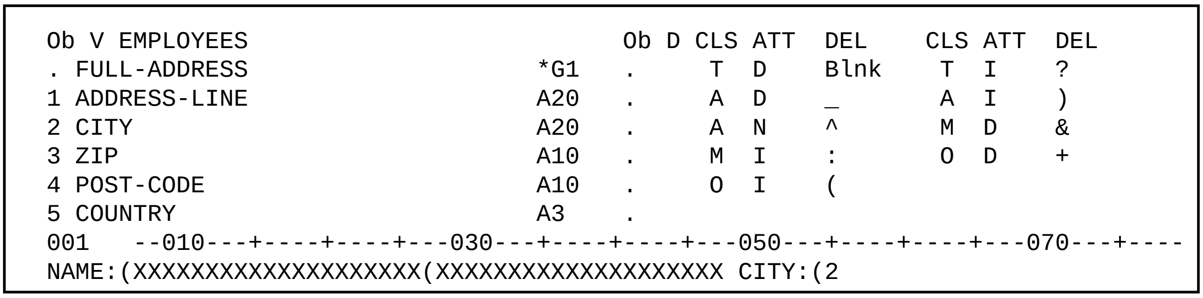
Two fields are now defined on the map using the field definitions of the fields NAME and FIRST-NAME taken from the view. When you press ENTER, the screen will look as below:



```
Ob V EMPLOYEES                          Ob D CLS ATT  DEL    CLS ATT  DEL
1 PERSONNEL-ID                    A8    .     T  D    Blnk    T  I    ?
. FULL-NAME                       *G1   .
2 FIRST-NAME                      A20   .
3 MIDDLE-I                        A1    .                     O  D    +
4 NAME                            A20   .     O  I    (
5 MIDDLE-NAME                     A20   .
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
NAME:(XXXXXXXXXXXXXXXXXXXX
(XXXXXXXXXXXXXXXXXXXX
```

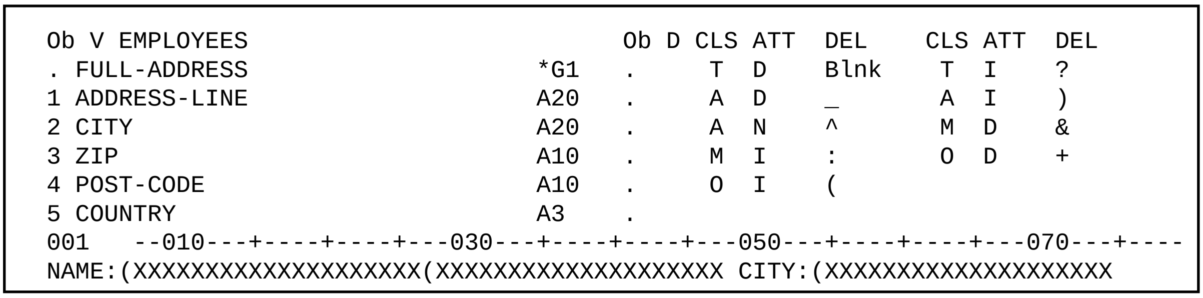
On the top line of the screen, type in a plus (+) sign over the "V". Repeat this step until the field "2 CITY" appears in the list. Use the minus (-) sign if you want to scroll backward.

Use the command ".M" to move the field from the second line of the editing area to the position shown below. Then enter "CITY:(2" as shown below:



```
Ob V EMPLOYEES                          Ob D CLS ATT  DEL    CLS ATT  DEL
. FULL-ADDRESS                    *G1   .     T  D    Blnk    T  I    ?
1 ADDRESS-LINE                    A20   .     A  D    _       A  I    )
2 CITY                            A20   .     A  N    ^       M  D    &
3 ZIP                             A10   .     M  I    :       O  D    +
4 POST-CODE                       A10   .     O  I    (
5 COUNTRY                         A3    .
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
NAME:(XXXXXXXXXXXXXXXXXXXX(XXXXXXXXXXXXXXXXXXXX CITY:(2
```

The map should look as follows:

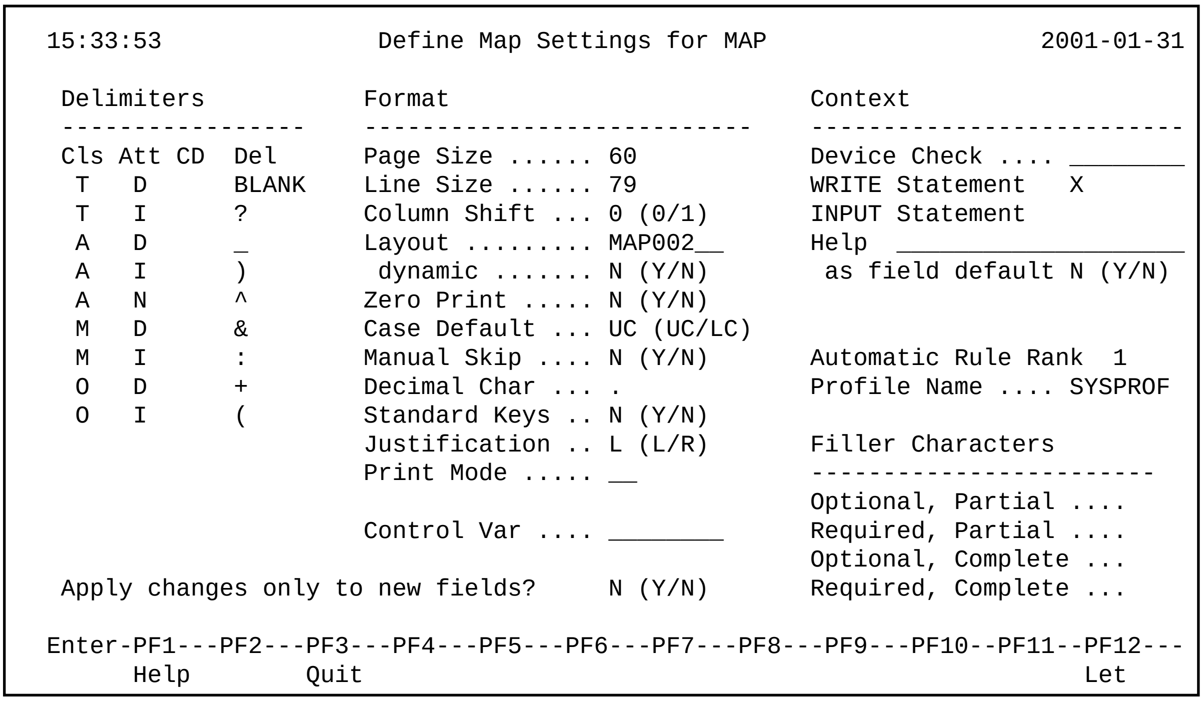


```
Ob V EMPLOYEES                          Ob D CLS ATT  DEL    CLS ATT  DEL
. FULL-ADDRESS                    *G1   .     T  D    Blnk    T  I    ?
1 ADDRESS-LINE                    A20   .     A  D    _       A  I    )
2 CITY                            A20   .     A  N    ^       M  D    &
3 ZIP                             A10   .     M  I    :       O  D    +
4 POST-CODE                       A10   .     O  I    (
5 COUNTRY                         A3    .
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
NAME:(XXXXXXXXXXXXXXXXXXXX(XXXXXXXXXXXXXXXXXXXX CITY:(XXXXXXXXXXXXXXXXXXXX
```

Press PF3 to return to the Edit Map menu.

Enter the code "W" to stow map MAP002.

Enter the code "I" and name MAP003. The Define Map Settings For Map screen will be displayed. Change the Page Size to 60; mark "WRITE Statement" with an "X"; unmark "INPUT Statement"; and type in MAP002 after "Layout". The map settings should be as follows:



```
15:33:53               Define Map Settings for MAP                   2001-01-31

 Delimiters            Format                         Context
 -----------------     ---------------------------    --------------------------
 Cls Att CD  Del       Page Size ...... 60            Device Check .... ________
  T   D      BLANK     Line Size ...... 79            WRITE Statement   X
  T   I      ?         Column Shift ... 0 (0/1)       INPUT Statement
  A   D      _         Layout ......... MAP002__      Help  ____________________
  A   I      )          dynamic ....... N (Y/N)        as field default N (Y/N)
  A   N      ^         Zero Print ..... N (Y/N)
  M   D      &         Case Default ... UC (UC/LC)
  M   I      :         Manual Skip .... N (Y/N)       Automatic Rule Rank  1
  O   D      +         Decimal Char ... .             Profile Name .... SYSPROF
  O   I      (         Standard Keys .. N (Y/N)
                       Justification .. L (L/R)       Filler Characters
                       Print Mode ..... __            ------------------------
                                                      Optional, Partial ....
                       Control Var .... ________      Required, Partial ....
                                                      Optional, Complete ...
 Apply changes only to new fields?     N (Y/N)        Required, Complete ...

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help        Quit                                                    Let
```

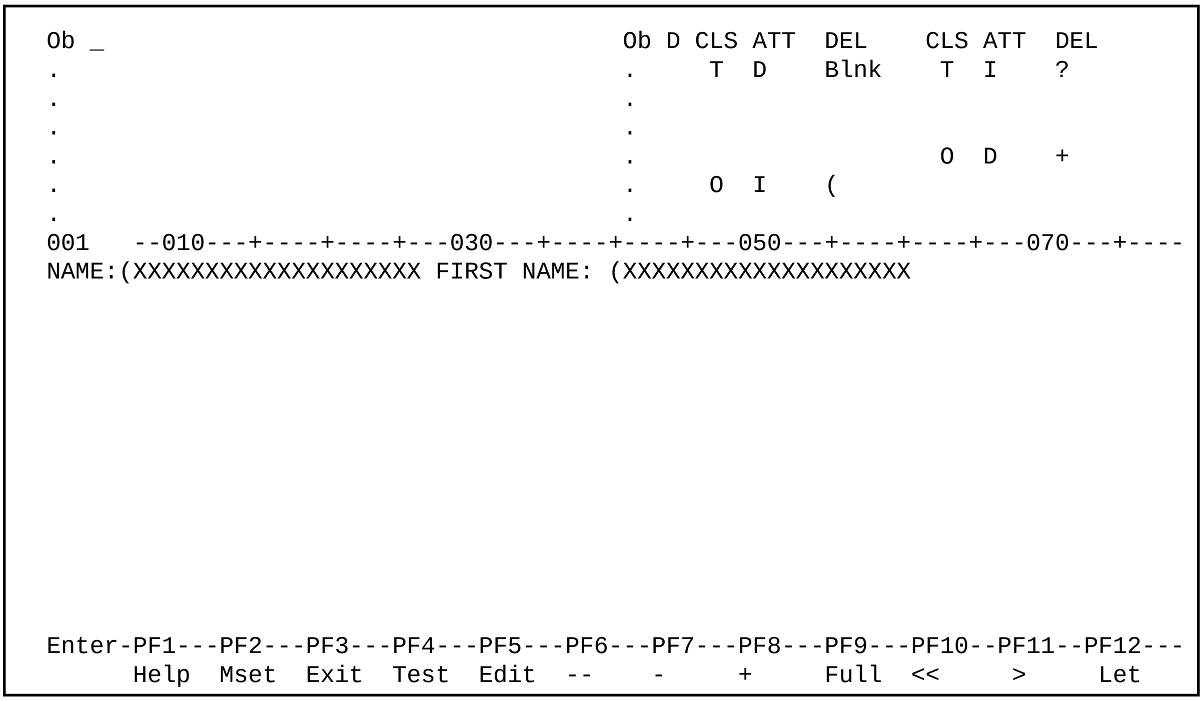
When you press ENTER, the map editing screen will be displayed with the layout of map MAP002 in the edit area.

Use the command ".T" to delete CITY:(XXXXXXXXXXXXXXXXXXXX.

Use the command ".M" to move the second of the remaining output fields to the right.

Insert the text "FIRST NAME:" into the line.

The map should now look as shown below:



```
Ob _                                    Ob D CLS ATT  DEL    CLS ATT  DEL
.                                       .     T  D    Blnk    T  I    ?
.                                       .
.                                       .
.                                       .                     O  D    +
.                                       .     O  I    (
.                                       .
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
NAME:(XXXXXXXXXXXXXXXXXXXX FIRST NAME: (XXXXXXXXXXXXXXXXXXXX









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test  Edit  --    -     +     Full  <<     >     Let
```

Press PF3 to return to the Edit Map menu.

Enter the code "W" to stow map MAP003.

In the Command line, enter EDIT PROG002.

The program editor will appear with program PROG002 in the work area.

RUN the program. Map MAP001 will be displayed.

Enter the name JONES and no city. The list produced by the program will now use MAP002:

```
Page      1                                                2001-01-31  15:38:11

NAME: JONES                VIRGINIA            CITY: TULSA
NAME: JONES                MARSHA              CITY: MOBILE
NAME: JONES                ROBERT              CITY: MILWAUKEE
NAME: JONES                LILLY               CITY: BEVERLEY HILLS
NAME: JONES                EDWARD              CITY: CAMDEN
NAME: JONES                MARTHA              CITY: KALAMAZOO
NAME: JONES                LAUREL              CITY: BALTIMORE
NAME: JONES                KEVIN               CITY: DERBY
NAME: JONES                GREGORY             CITY: NOTTINGHAM
```

Press ENTER to return to MAP001.

Enter the name JONES and the city DERBY. Map MAP003 will be displayed:

```
Page      2                                                2001-01-31  15:39:11

THE FOLLOWING EMPLOYEES LIVE IN  DERBY
NAME: JONES                FIRST NAME:  KEVIN
```

Press ENTER again to return to MAP001.

Enter a period (.) in the NAME field to return to the program. STOW the program.

End of Session 5.

Program Editor

The Natural program editor is used to perform online full-screen editing of Natural source programs. With the program editor, you can create and edit source programs quickly and efficiently with a minimum of effort. This section describes how to use this editor:

- Invoking the Program Editor
- Top Information Line
- Bottom Information Line
- Editor Command Line
- Editing a Program
- Editor Commands
- Editor Commands for Positioning
- Line Commands
- Special PF-Key Functions
- Cursor-Sensitive Commands
- The Exit Function

Invoking the Program Editor

You invoke the program editor with the system command EDIT as described in the Natural System Command Reference documentation.

When you invoke the program editor, the editor screen is displayed (as shown below with a program in the work area):

```
>                                       > +  Program     SAGDEMO  Lib SAGTEST
 All   ....+....1....+....2....+....3....+....4....+....5....+.Mode Structured.
   0010 DEFINE DATA LOCAL USING L-INVOIC
   0020             LOCAL USING L-INV-LN
   0030 END-DEFINE
   0040 *
   0050 READ INVOICE-VIEW BY INVOICE-NO FROM 1
   0060 *
   0070   FIND INVOICE-LINE-VIEW WITH INVOICE-NO = INVOICE-NO (0050)
   0080     DELETE
   0090   END-FIND
   0100 *
   0110   DELETE
   0120   END TRANSACTION
   0130 END-READ
   0140 *
   0150 FETCH 'MENU'
   0160 END
   0170
   0180
   0190
   0200
         ....+....Current Source Size: 308 Char. Free: 44756   +.. S 16   L 1
```

Note:
If Natural ISPF is installed and the general editor profile default "ISPF Editor as Program Editor" is set to "Y", instead of the program editor, either the Natural ISPF main menu (if the EDIT command is entered without an object name) or the Natural ISPF editor screen with the specified object is invoked.

Top Information Line

The top information line of the editor screen is used to display a message indicating object modification. See also the section Source Status Message. In addition, the programming mode (structured or reporting) currently in effect is displayed. When a program is read into the edit area, the mode is set to the one which was in effect when the program was stowed. This information is only displayed if the "Source Size Information" parameter in the editor profile defaults is set to "Y".

Bottom Information Line

In the bottom information line of the editor screen, the following items of information are displayed:

Current Source Size	Size (number of characters) of the current object. As source lines are stored in variable length in the work area, trailing blanks within a source line are not counted; leading and embedded blanks are counted. This information is only displayed if the "Source Size Information" parameter in the editor profile defaults is set to "Y".
Char. Free	The number of characters still available in the work area. This information is only displayed if the "Source Size Information" parameter in the profile defaults is set to "Y".
S	Size (number of lines) of the object being edited.
L	The number of the source line currently displayed as the top line.

Editor Command Line

The top line of the program editor screen is the edit command line. In this line, you can enter:

- a Natural system command (for example, EDIT, CHECK, SAVE),
- one or more editor commands,
- the name of a Natural program to be executed.

Additionally, the following items of information are displayed:

Direction Indicator (+ or -)	The direction indicator can be set to control the direction of the editor commands ADD and SCAN and of the line commands ".C", ".I" and ".M". The value "+" indicates **after** and the value "-" indicates **before**. The exact interpretation is described with the relevant command description.
Object Type	The type of object currently in the work area. The object type can be changed by using the editor command SET TYPE.
Object Name	The name of the object currently in the work area.
Library (Lib)	The library to which you are currently logged on.

Editing a Program

Multiple Functions

Multiple functions can be performed from a single input screen:

- Source lines can be updated directly.
- One or more line commands can be used.
- One or more editor commands can be used.

The following restrictions related to multiple functions apply:

- Only one insert line command (.I) can be performed at a time.
- You can enter multiple commands in the command line of the editor: you can enter more than one editor command, but only the last command entered in the editor command line can be a Natural system command. For example:
  SC ’MOVE’,-2,RENUMBER.
- If you have changed the screen contents manually or by commands, a system command cannot be entered until you press ENTER.

Note:
Natural treats the editor command "N" like a system command.

Dynamic Conversion from Lower to Upper Case

When the Natural terminal command %L is set and dynamic conversion to upper case is specified, all source code you enter in the editor is automatically converted to upper case, with the following exceptions:

- Text strings that are not hexadecimal constants and are enclosed in apostrophes remain as you enter them.
- Text strings (with or without apostrophes) in objects of type Text remain as you enter them.
- Comments remain as you enter them.

Dynamic conversion from lower to upper case can be specified and deactivated in the editor profile.

Editor Commands

Editor commands are entered in the command line of the program editor. The command parameters must be separated either by the input delimiter character as defined with the Natural session parameter ID (the default delimiter character is comma ",") or by a blank. When multiple commands are entered, these must also be separated by the delimiter character or by blanks. Line commands must not be entered in the command line.

The following edit commands are available:

Editor Command	Function
ADD[(*n*)]	This command adds *n* blank lines. If the direction indicator is set to "+", the lines are added after the last line of the object being edited; if the direction indicator is set to "-", the lines are added before the first line of the object. The value for *n* can be in the range from 1 to 9. If *n* is not (or not correctly) specified, 9 lines (4 in split-screen mode) are added by default. With the next ENTER, lines that are still left blank will be eliminated.
CANCEL	With this command you leave the editor. Any modifications made since the last time the SAVE command was entered are **not** saved.
CHANGE	This command scans for the value entered as *scan-value* and replaces each such value found with the value entered as *replace-value*. The syntax for this command is: CHANGE ’*scan-value* **’***replace-value* **’** Any special character which is not valid within a Natural variable name can be used as the delimiter character.
CLEAR	This command clears the edit area (including the line markers "X" and "Y").
DX, DY, DX-Y	This command deletes the X-marked line; or the Y-marked line; or the block of lines delimited by "X" and "Y". See also the line commands ".X" and ".Y".
EX, EY, EX-Y	This command deletes source lines from the top of the source area to, but not including, the X-marked line; or from the source line following the Y-marked line to the bottom of the source area; or all source lines in the source area excluding the block delimited by "X" and "Y". See also the line commands ".X" and ".Y".
EXIT	With this command you leave the editor.
LET	This command undoes all modifications made to the current screen since the last time ENTER was pressed. In addition, LET ignores all line commands already entered but not yet executed.
N [(*n*)]	This command renumbers the source code lines of the program currently in the work area. If you only enter "N", the lines are numbered in increments of 10; if you enter "N (*n*)", the lines are renumbered in increments of *n*. If the value specified for *"n"* is too big, lines are numbered in increments of 5. **Note:** See also Renumbering of Source-Code Line Number References.
PROFILE [*name*]	This command displays the current editor profile.
QUIT	Same as editor command CANCEL.

Editor Command	Function
REN ON|OFF	ON Renumbers a Natural source program whenever it is checked, run, saved, stowed or cataloged. OFF Indicates that automatic renumbering is not in effect. The default is ON (see also the section Editor Profile). **Note:** See also Renumbering of Source-Code Line Number References.
RESET	This command deletes the current X and Y line markers and any marker previously set with the line command ".N". See also line commands ".X" and ".Y".
SCAN [’*scan-value*’]	This command scans for data in the source area. If you enter SCAN without any parameter, the SCAN menu is invoked. If you enter SCAN ’*scan-value*’, a scan for *scan-value* is performed. If the supplied scan value is entered without delimiter characters, for example, "SCAN ABC D", the entire character string which follows the keyword SCAN is used as the scan value. SCAN is a cursor-sensitive command.
SCAN = [+|-]	This command scans for the next occurrence of the scan value. The direction of the scan operation is determined by the setting of the direction indicator. If the direction indicator is omitted or set to "+", the scan operation will be from the current position of the source area (top of the displayed source window) to the last line in the source area. If the direction indicator is set to "-", the scan operation will be backwards from the bottom line of the current screen to the first line in the source area. The direction for a given scan command can also be explicitly specified by entering "SCAN =+" or "SCAN =-" prior to command execution. The first line which contains the scanned value is positioned to the top line after each SCAN command. Each line in which the scanned value is located is marked with an "S" to the left of the line. **Note:** The equal sign "=" used with the SCAN command is the default input assign character. If another character has been specified as input assign character (see session parameter IA as described in the Natural Parameter Reference documentation), that other character must be used instead.
SET ABS [ON|OFF]	ON The SCAN and CHANGE commands operate in absolute mode, which means that the value to be scanned/changed need not be delimited by blanks or special characters. OFF The SCAN and CHANGE commands do not operate in absolute mode, which means that the value to be scanned/changed must be delimited by blanks or special characters. The default is OFF.
SET ESCAPE *character*	The escape character which must precede each line command. The default escape character is ".".

Editor Command	Function
SET NUL [ON|OFF]	ON All occurrences of a value scanned with the SCAN command are deleted. After the deletion of the scanned value, the SET NUL command is automatically set to OFF. The default is OFF.
SET RANGE [ON|OFF]	ON The SCAN and CHANGE commands operate in range mode, which means that the value to be scanned/changed must be located within the range of lines delimited by the X and Y line markers. OFF The SCAN and CHANGE commands operate in non-range mode, which means that no range limit is to be in effect. The default is OFF.
SET SEQ [ON|OFF]	OFF If your input is numeric, the first four positions in the edit area are considered as the line number and are moved to the line number position once you press ENTER. This feature is useful, for example, if a statement line is to be referenced by a source code line number in another statement line; when you renumber the source code, the referencing line number is renumbered, too. ON Numeric input in the first four positions remains as entered. Except with object type Text, the default is OFF.
SET SIZE [ON|OFF]	ON The program size is displayed at the bottom information line of the editor screen and the programming mode is displayed on the scale line. OFF This information is not displayed. The default is OFF.
SET STAY [ON|OFF]	ON The current screen will stay when ENTER is pressed. Forward and backward positioning can be done by positioning commands only. OFF Pressing ENTER positions to the next screen.

SET TYPE	The object type is set automatically when an existing object is read into the work area.
	This command can be used to change the type of object to be edited:
	SET TYPE PROGRAM (Natural Program)
	SET TYPE SUBROUTINE (Natural Subroutine)
	SET TYPE SUBPROGRAM (Natural Subprogram)
	SET TYPE HELPROUTINE (Natural Helproutine)
	SET TYPE COPYCODE (Natural Copycode)
	SET TYPE TEXT (Natural Text)
	SET TYPE CLASS (Natural Class)

SHIFT [-|+*nn*]	This command shifts each source line delimited by the X and Y markers to the left or right. The *nn* parameter represents the number of characters the source line is to be shifted. Comment lines are not shifted.
SHIFT - -	This command shifts each source line delimited by the X and Y markers to the leftmost position. Comment lines are not shifted.

Editor Command	Function
SHIFT ++	This command shifts each source line delimited by the X and Y markers to the rightmost position (maximum 99 positions). Comment lines are not shifted.
STRUCT [DISPLAY]	This command performs structural indentation of a Natural source program. If DISPLAY is specified, the Natural source program is displayed in compressed form (see also the system command STRUCT in the Natural System Command Reference documentation).
*	This command displays the editor command most recently entered.
*=	This command again executes the command most recently entered in the command line.
.	With this command you leave the editor. Any modifications made since the last time the SAVE command was entered are **not** saved.

See also Renumbering of Source-Code Line Number References in the Natural Reference Documentation.

Editor Commands for Positioning

Editor commands for positioning are entered in the command line of the program editor. The following commands are available for positioning:

Command	Function
+P	Position forwards one page.
+	
-P	Position backwards one page.
-	
+H	Position forwards half a page.
-H	Position backwards half a page.
T	Position to top of program.
- -	
B	Position to bottom of program.
++	
+*nnnn*	Position forwards *nnnn* lines (maximum 4 digits).
-*nnnn*	Position backwards *nnnn* lines (maximum 4 digits).
nnnn	Position to line number *nnnn*.
X	Position to the line marked with "X".
Y	Position to the line marked with "Y".
POINT	Positions to the line in which the line command ".N" was entered. See also the line command ".P".

Line Commands

The line commands are listed below. The notation "(*nnnn*)" indicates a repetition factor. The default repetition value is 1 (with the exception of the ".I" command; see below).

Note:
You are recommended to enter a blank at the end of each line command. This prevents the editor from attempting to interpret any information existing on the line as part of the line command.

Line Command	Function
.C(*nnnn*)	Copies the line in which the command was entered.

.CX(*nnnn*)
.CY(*nnnn*)
	Copies the X-marked or the Y-marked line. See also the line commands ".X" and ".Y" as well as the notes in the following section.

.CX-Y(*nnnn*)	Copies the block of lines delimited by the X and Y markers. (See also the notes in the following section.)
.D(*nnnn*)	Deletes line or lines. The default is 1 line.
.I(*n*)	Inserts *n* empty lines, where *n* can be in the range from 1 to 9. If *n* is not (or not correctly) specified, 9 lines (4 lines in split-screen mode) are inserted by default. (See also the notes in the following section.)
.I(*obj*,*ssss*,*nnnn*)	Includes into the source an object contained in the current library or in the steplib (the default steplib is SYSTEM). Depending on the direction indicator, the object is inserted before or after the line in which you enter the command. If you wish to include only part of the object, you specify as *ssss* the first line to be included (e.g., "20" means the inclusion will start from the 20th line), and as *nnnn* the number of lines to be included. If you enter multiple commands, this command is always executed after all other line and/or editor commands have been executed. If the object is a map, an INPUT USING MAP statement with all defined variables is automatically included in the current line. If the object is a data area, the entire data area is included, except comment lines. Only stowed local and parameter data areas can be included into the source area; global data areas cannot be included.
.J	Joins the current line with the next line. If the resulting line exceeds the length of the editor screen line, the line is marked with "L" and must be split in two with the ".S" command (see below) before it can be modified.
.L	Undoes all modifications that have been made to the line since the last time ENTER was pressed.
.MX .MY	Moves the X-marked or the Y-marked line. See also the line commands ".X" and ".Y" as well as the notes below.
.MX-Y	Moves the block of lines delimited by the X and Y markers (see also the notes below).

Line Command	Function
.N	Marks (invisibly) a line to be positioned to the beginning of the source area by the editor command POINT. The mark is automatically deleted when an error with a line command or editor command occurs.
.P	Positions the line marked by this command to the top of the screen.
.S	Splits the line at the position marked by the cursor.
.X	Marks a line or the beginning of a block of lines, to be processed (see also the notes below).
.Y	Marks a line or the end of a block of lines, to be processed (see also the notes below).

Note:
If both the commands ".X" and ".Y" are applied to one line, it is treated as being marked with "X" and with "Y"; the line marker actually shown to reflect this status is a "Z".

If the direction indicator is set to "+", the copied, inserted or moved lines are placed after the line in which the corresponding command was entered; if the direction indicator is set to "-", the copied, inserted or moved lines are placed before the line in which the command was entered.

Special PF-Key Functions

The following special functions can also be controlled using PF keys:

Function	Explanation
*CURSOR	A line split function can be combined with the command ".I", ".CX", ".CX-Y", ".MX" or ".MX-Y". This is accomplished by assigning the value "*CURSOR" to a PF key. If this PF key is then pressed instead of ENTER after a line command has been entered, the line in which the command was entered is first split at the cursor position and then the line command is executed.
*X *Y	If a PF key is assigned the value "*X" or "*Y", the cursor position is marked X or Y whenever this PF key is used. These column markers are then used to determine which portion of a line is to be included in the command operation. See the example below.

Example:

```
                  XY
X       0010 MOVE A TO B
        0020 WRITE A B
Y       0030 MOVE B TO A
                  XY
        ....
        0100 .MX-Y.....
```

The block of text starting with the "A" in line 0010 and ending with the "B" in line 0030 is moved:

```
0010 MOVE
0030 TO A
....
0010 A TO B
0020 WRITE A B
0030 MOVE B
```

Cursor-Sensitive Commands

- The SCAN Commands
- The SPLIT Command
- The EDIT and LIST System Commands

Cursor-sensitive commands are commands where, instead of entering a name in the command line, you can mark the name with the cursor anywhere on the editor screen (except in the command line). You can place the cursor on any word that is not in the command line. It does not matter where on the word the cursor is placed.

The SCAN Commands

The SCAN [’*scan-value*’] command scans for data in the edit area. If the SCAN command is used without any parameter but with the cursor positioned outside the editor command line, this results in a scan operation for the string the cursor is positioned to. If the cursor is positioned to a blank character, however, the SCAN menu is invoked.

In split-screen mode, the cursor can be positioned to a string in the split-screen area, too. The scan operation, however, is performed in the edit area only.

When using the SPLIT SCAN [’*scan-value*’] command, the same applies as for the SCAN command, but the scan operation is performed in the split-screen area only (see also the section Split-Screen Commands).

Note:
To benefit from cursor sensitiveness as much as possible, the SCAN or SPLIT SCAN command should be assigned to a PF key.

The SPLIT Command

Instead of the commands SPLIT PROGRAM, SPLIT DATA, SPLIT FUNCTION and SPLIT VIEW, which you can use to display a programming object or DDM in the split-screen area of the editor (see also the section Split-Screen Commands), you only have to enter the command SPLIT and place the cursor on the name of the desired object. The object must be contained in the current library.

Note:
To benefit from cursor sensitiveness as much as possible, the SPLIT command should be assigned to a PF key.

The EDIT and LIST System Commands

The system commands EDIT and LIST are cursor-sensitive, too. Instead of specifying an object name, the cursor can be positioned to a text string of the object currently in the edit area that corresponds to the desired object name.

With the EDIT command, the corresponding object is loaded into the editor. If necessary, even a different editor is invoked.

With the LIST command, the corresponding object is listed, even if a view has been referenced.

For more information on EDIT and LIST see the Natural System Command Reference documentation.

The Exit Function

If the editor default parameter "Prompt Window for Exit Function" is set to "Y", any time you enter the EXIT command in the command line, the EXIT Function prompt window is invoked, offering you the following options:

Option	Explanation
Save and Exit	Leaves the editor and saves all modifications made to the current object.
Exit without Saving	Leaves the editor without saving any modification made to the current object since the last SAVE command was entered.
Resume Function	Neither leaves the editor nor saves any modifications; the prompt window is closed and the current function is resumed.

When "Prompt Window for Exit Function" is set to "N", the EXIT command leaves the editor and saves all modifications made to the current object; no prompt window is displayed.

Data Area Editor

The Natural data area editor is used to define and maintain definitions for global, local and parameter data areas.

A data area definition can consist of user-defined variables, database views and global data blocks (a collection of variables and/or views).

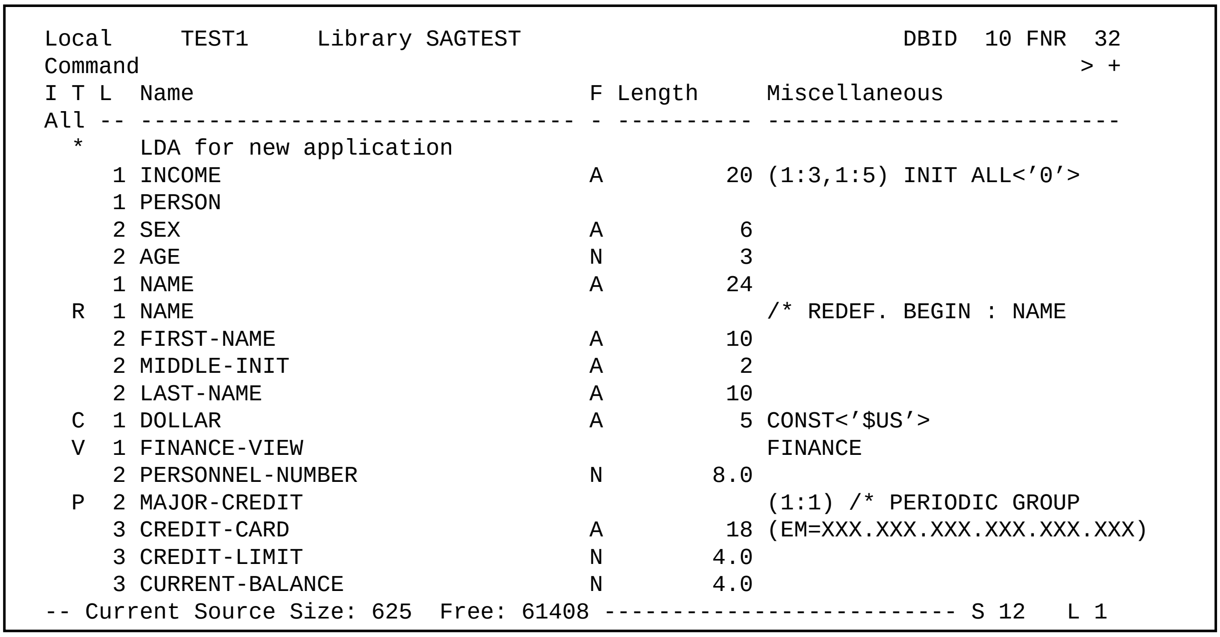
This section covers the following topics:

- Invoking the Data Area Editor
- Top Information Line
- Bottom Information Line
- Editor Command Line
- Editing a Data Area
- Editor Commands
- Line Commands
- Exit Function
- Sharing Data Area Sources by Natural Version 3.1 and 4.1

Invoking the Data Area Editor

You invoke the data area editor with the system command EDIT, specifying a data area type (GLOBAL, LOCAL or PARAMETER) or the name of a data area with the command (for details, see the system command EDIT as described in the Natural System Command Reference documentation). If you specify the name of a data area, it is read into the edit area of the data area editor.

The data area editor screen appears with a local data area in the edit area:



```
Local     TEST1     Library SAGTEST                            DBID  10 FNR  32
Command                                                                     > +
I T L  Name                             F Length     Miscellaneous
All -- -------------------------------- - ---------- --------------------------
  *    LDA for new application
     1 INCOME                           A         20 (1:3,1:5) INIT ALL<’0’>
     1 PERSON
     2 SEX                              A          6
     2 AGE                              N          3
     1 NAME                             A         24
  R  1 NAME                                          /* REDEF. BEGIN : NAME
     2 FIRST-NAME                       A         10
     2 MIDDLE-INIT                      A          2
     2 LAST-NAME                        A         10
  C  1 DOLLAR                           A          5 CONST<’$US’>
  V  1 FINANCE-VIEW                                  FINANCE
     2 PERSONNEL-NUMBER                 N        8.0
  P  2 MAJOR-CREDIT                                  (1:1) /* PERIODIC GROUP
     3 CREDIT-CARD                      A         18 (EM=XXX.XXX.XXX.XXX.XXX.XXX)
     3 CREDIT-LIMIT                     N        4.0
     3 CURRENT-BALANCE                  N        4.0
-- Current Source Size: 625  Free: 61408 ------------------------- S 12   L 1
```

Top Information Line

The top information line of the editor screen displays the type and name of the data area currently in the editor, as well as the library, database ID and file number to which you are currently logged on.

Bottom Information Line

In the bottom information line of the editor screen, the following information is displayed:

Current Source Size	Size (number of characters) of the current object. As source lines are stored in variable length in the work area, trailing blanks within a source line are not counted; leading and embedded blanks are counted. This information is only displayed if the "Source Size Information" parameter in the editor profile defaults is set to "Y".
Free	The number of characters still available in the work area. This information is only displayed if the "Source Size Information" parameter in the profile defaults is set to "Y".
S	Size (number of lines) of the object being edited.
L	The number of the source line currently displayed as the top line.

Editor Command Line

The second line of the data editor screen is the edit command line. In this line, you can enter:

- a Natural system command (for example, EDIT, CHECK, SAVE),
- one or more editor commands,
- the name of a Natural data area to be executed.

In addition, the direction indicator can be set to control the direction of several editor and line commands. The value "+" indicates **after** and the value "-" indicates **before**. The exact interpretation is described with the relevant command description.

Editing a Data Area

The editor screen of the data area editor is divided into columns of fields with the following possible entries:

Field	Explanation
I	Label Indicator. Information field supplied by the editor. This column is not modifiable by the user. + indicates that more than one of the entries listed below has been defined using the ".E" line command. Possible entries are: E indicates that a definition error has been detected. A indicates that array bounds have been defined using the ".E" line command. I indicates that an initial value has been defined using the ".E" line command. M indicates that an edit mask has been defined using the ".E" line command. S indicates that both an initial value and an edit mask have been defined using the ".E" line command. Parameter Data Areas only (see also Extended Field Definition Editing): blank indicates the parameter specification BY REFERENCE (default). V indicates the parameter specification BY VALUE. R indicates the parameter specification BY VALUE RESULT. O indicates the parameter specification OPTIONAL.
T	Type. Possible types are: B Data block C Constant (user-defined variable only) or Counter field (database field only) * Comment G Group (within a view only) M Multiple-value field O Handle of object P Periodic group R Redefinition U Globally Unique Identifier (GUID) V View
L	Level number (1 - 99). Variables which are not within a hierarchical structure must be assigned level 1. View definitions must be assigned level 1. Level numbers cannot be used with data block definitions.
Name	Name of the variable, block or view. Instead of specifying a variable name, the filler option (*n*X) can be used. With the filler option, *"n"* filler bytes can be denoted within a field or variable being redefined, where *"n"* can be up to 10 digits (smaller than 1 GB). The definition of trailing filler bytes is optional.
F	Format. Any format supported by Natural can be used.

Field	Explanation
Length	Length. No length is permitted for formats C, D, T and L. You can define dynamic variables by specifying "DYNAMIC" in the length field.
Miscellaneous	This field can be used to define the bounds for an array, to supply initial values for a variable or to supply an edit mask for a variable; for a view definition, the name of a DDM from which this view is derived must be entered; for a block definition, the name of the parent block must be entered; and a comment can be entered. See also the examples below. Together with an edit mask, also a field header (HD) and the print mode (PM) can be defined: (HD=’Name’ EM=XXX.XXX.XX PM=N) See the Natural Parameter Reference documentation for further information on the session parameter PM . As this field may be too short to make all necessary or desired specifications, the ".E" line command is provided for extended field definition editing. **Note:** When defining a view, the name of the DDM from which this view is derived can be modified. However, this is only possible if all fields of the view are also contained in the DDM with the modified name.

Examples of Array Definitions:

```
(2,2)   (2 dimensions, 2 occurrences)
(2,2,2) (3 dimensions, 2 occurrences)
(1:10,2)
(-1:3,2)
```

Examples of Initial Value Assignments:

```
INIT<3>
INIT<’ABC’>
INIT<H’FF’>
CONST<12>
```

Example of an Edit Mask Definition:

```
(EM=999.99)
```

Editor Commands

The following editor commands can be entered in the command line of the data area editor:

Command	Function
CATALOG [*name*]	This command catalogs the data area definition currently in the edit area.
CHECK	This command checks the data area definition currently located in the edit area. It also orders the entries in the Miscellaneous column in the following sequence: - index - initial value - edit mask - name - comment
CLEAR	This command clears the edit area.
DX, DY, DX-Y	This command deletes a line marked with "X" or "Y", or a block of lines delimited by "X" or "Y".
EX, EY, EX-Y	This command deletes: ● source lines from the top of the source area to the line marked with "X", but not including this line; or ● source lines from the line marked with "Y", but not including this line, to the bottom of the source area; or ● all source lines in the source area, excluding the block delimited by "X" or "Y".
EXIT	With this command you leave the data area editor.
GENERATE [*name*]	This command generates Natural copycode using the data area definitions currently in the edit area. A DEFINE DATA LOCAL and corresponding END-DEFINE statement are automatically included. If a *name* is entered, the generated copycode is saved under this name.
M +|-	Scrolls the Miscellaneous field. + Scrolls to the right. - Scrolls to the left.
PROFILE [*name*]	This command displays the current editor profile.
READ *name*	This command reads an existing data area definition into the edit area.
SAVE	This command stores a source object. The contents of the source work area are not affected.
SET ABS [ON|OFF]	This command determines whether the SCAN command operates in absolute or non-absolute mode. ON: the SCAN command operates in absolute mode, which means that the value to be scanned need not be delimited by blanks or special characters. OFF: the SCAN command operates in non-absolute mode, which means that the value to be scanned must be delimited by blanks or special characters. The default is OFF.

Command	Function
SET PREFIX *prefix*|OFF	This command allows you to specify a prefix for field names. This prefix is then automatically placed before the value entered in the "Name" column for each line that is entered or modified, unless the name already begins with this prefix. If the concatenated variable is longer than 32 bytes, a message is given and the value in the name field can be shortened. If this is not done, the prefix will not be inserted.
SET SCAN COMMENT|NAME	If SET SCAN is set to COMMENT, you can scan for a value in the "Comment" column. If SET SCAN is set to NAME, you can scan for a value in the "Name" column. You cannot scan in both columns simultaneously; the default is NAME.
SET SIZE ON|OFF	If SET SIZE is set to ON, the size of the data area is displayed at the bottom information line of the editor screen.
SET STAY ON|OFF	If STAY is set to ON, the current screen will stay when ENTER is pressed. Forward and backward positioning can be done by positioning commands only. If STAY is set to OFF, pressing ENTER positions to the next screen.
SET TYPE G|L|P	This command sets the data area object type: G Global data area L Local data area P Parameter data area
STOW [*name*]	This command saves and catalogs the data area definition currently in the edit area.

Line Commands

All line commands described for the Natural program editor (except those which require a line number) can be used in the data area editor as well.

You are recommended to enter a blank at the end of each line command. This prevents the editor from attempting to interpret any information existing on the line as part of the line command.

In addition, the following line commands are available for the data area editor:

Command	Function
.D	This command deletes one or more lines. When entered for an individual field, only that field definition is deleted. When entered for a part of a hierarchical structure (view, group, redefinition), all subsequent definitions on subordinate levels are also deleted. If, for example, you enter ".D" for a group defined at level 2, everything belonging to that group and with a level number greater than 2 is also deleted up to (but not including) the next level 2 definition. Comment lines (which usually are not assigned a level) are also considered to be at a subordinate level. To avoid the undesired deletion of a comment, assign an appropriate level to it. **Note:** In the data area editor, the ".D" command works differently from the program editor.
.D(*nnnn*)	This command deletes *nnnn* lines, beginning with the line in which you enter the command. Unlike ".D" (see above), ".D(*nnnn*)" affects only the number of lines specified, regardless of any hierarchical structure.
.E	Using this command you can define array bounds, initial values, edit masks and parameter attributes. For more information, see the section Extended Field Definition Editing below. If ".E" is executed for a DDM field, the Define Edit Mask / Header screen is invoked immediately, because only edit masks and headers can be defined for DDM fields. It is not possible to define initial values for DDM fields.
.F(*file-name*)	This command includes a Predict file (applicable to file types: Conceptual, Standard, Sequential, Other).
.I(*n*)	This command adds *n* empty lines, where *n* can be in the range from 1 to 9. If *n* is not (or not correctly) specified, 10 lines (5 lines in split-screen mode) are added by default. If the direction indicator is set to "+", the lines are added after the current line of the object being edited; if the direction indicator is set to "-", the lines are inserted before the current line.

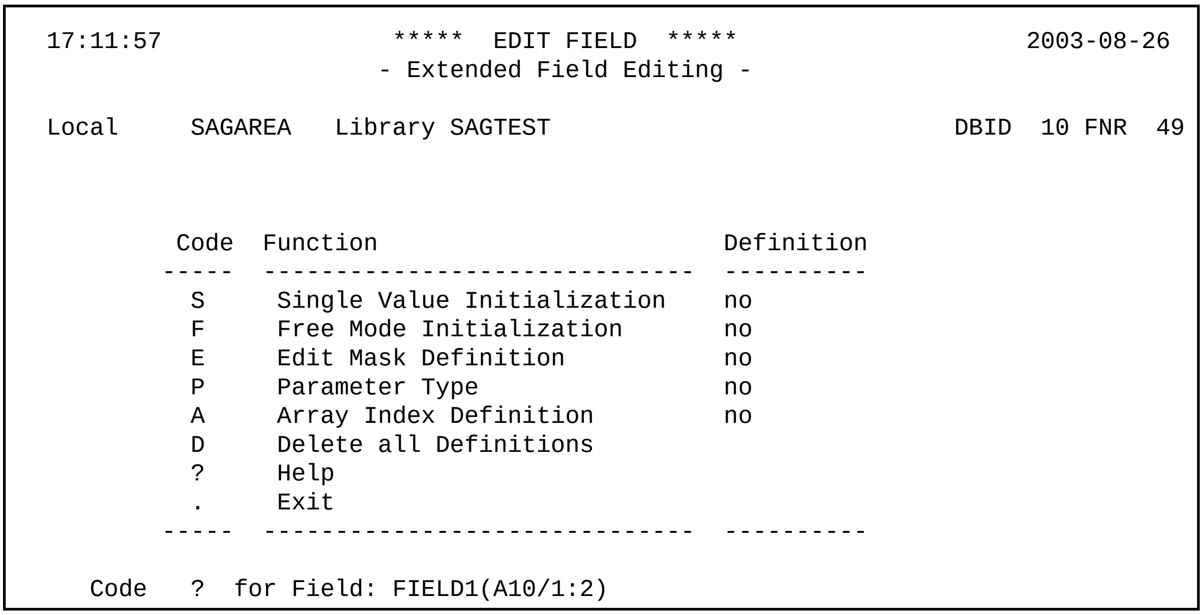
Command	Function
.I(*obj*)	This command includes a Natural object. Apart from data areas, the following object types can be specified: programs, subprograms, subroutines, helproutines, maps. If the object specified as *obj* is not a data area, it must be available as cataloged object. A window appears in the data area editor screen where you can select one of the following data definitions to be incorporated into your current data area: - all local variables and parameters contained in the specified object (including those incorporated from local and/or parameter data areas), - all local variables contained in the specified object (including those incorporated from local data areas), - only those local variables defined within the specified object, - all parameters contained in the specified object (including those incorporated from parameter data areas), - only those parameters defined within the specified object. If you incorporate variable definitions from objects without a DEFINE DATA definition (that is, from objects coded in reporting mode), variable redefinitions (see the REDEFINE statement in the Natural Statements documentation) might be placed to a wrong position; that is, after the wrong variable. So, before compiling your new data area, check all variable definitions and redefinitions for correct positioning. If a variable redefinition results in more than one variable, each variable is incorporated as one individual redefinition using filler bytes where appropriate. If the specified object has been cataloged using the Natural Optimizer Compiler, initial values and constants cannot be incorporated. If the object you want to insert has features the data area editor does not support, a message will be displayed and the relevant line will be marked as a comment line.
.I(*obj*,*ssss*,*nnnn*)	This command includes a global, local or parameter data area. This feature is only supported for data areas which do not contain initial values or edit masks. The "*ssss*" entry can be used to indicate at which line the insertion is to begin. For example, when setting "*ssss"* to 20, the insertion begins with the 20th line of the data area. The "*nnnn*" entry can be used to indicate the number of lines to be inserted. If *"ssss"* and/or *"nnnn" i*s specified for an object other than a data area (see the .I(obj) command), the specified value(s) are ignored.
.R	This command redefines a field or variable. With the filler option (*n*X), *n* filler bytes can be denoted within a field or variable being redefined. The definition of trailing filler bytes is optional.

Command	Function
.V [(*view-name*[,NOFL])]	This command defines a view. A view (DDM) layout is displayed. You then select the fields from the view which are to be used in the program. If no view name is specified, the view currently in the split screen is included. If ".V *view-name*" is specified within a view of the same name as specified for *view-name*, the selected fields are included in this view and no new view is defined. If NOFL is specified, the selected fields are included without format and length specification. When a periodic group or multiple-value field defined - in a DDM generated with Predict - as "PC" or "MC" respectively is included in a data area, a C* variable (internal count of occurrences) for the group or field is automatically generated and placed before the group or field. The index for such a periodic group or multiple-value field is defined with the number of occurrences defined in Predict. If the number of occurrences has not been defined in Predict, the maximum occurrences (191) are used. If Predict is active, Predict redefinitions and comments are incorporated, too. **Note:** With VSAM views, always the actual number of occurrences is displayed. In addition, VSAM views contain information on subdescriptors and superdescriptors (for further information, see the Natural for VSAM documentation.
.*	This command generates a C* variable for multiple-value fields or fields within a periodic group.
number [(*nnn*,*m*)]	This command is available in split-screen mode and with a view in the split-screen area only. To obtain fields and groups from the split-screen area, the level number of the field or group from the split-screen area must be specified in the first column (without a period "."). The field or group is inserted before or after the current line, depending on the setting of the direction indicator ("+" or "-"). Fields and groups from the split-screen area can be included as fields of a view (if *number* is entered inside a view) or as user variables. If the selected field has the same name as the field for which the command was entered, it is substituted instead of inserted. Multiple lines can be obtained from the split screen using the "*nnn*" notation where *nnn* is the number of lines to be included. The "*m*" notation can be used to specify a level number to be assigned to the field or group to be inserted.

Note:
".I(obj.)", ".R" and ".*" are available in full-screen mode only, not in split-screen mode.

Extended Field Definition Editing

To invoke Extended Field Definition Editing, enter the ".E" line command in front of a specific field. The following screen will be displayed:



```
17:11:57                *****  EDIT FIELD  *****                    2003-08-26
                       - Extended Field Editing -

Local     SAGAREA   Library SAGTEST                            DBID  10 FNR  49



         Code  Function                        Definition
        -----  ------------------------------  ----------
          S     Single Value Initialization    no
          F     Free Mode Initialization       no
          E     Edit Mask Definition           no
          P     Parameter Type                 no
          A     Array Index Definition         no
          D     Delete all Definitions
          ?     Help
          .     Exit
        -----  ------------------------------  ----------

   Code   ?  for Field: FIELD1(A10/1:2)
```

If any initial values or edit masks have been defined, the corresponding status message in the Definition column is changed from "no" to "yes".

The following function codes are available:

Code	Function
S	This function enables you to define an initial value for the specified field. You need only enter the desired field value; any further specifications necessary (including apostrophes for alphanumeric fields) are generated automatically. For an array (multiple-value field), an initial value can (but does not necessarily have to) be defined for each occurrence. With arrays, asterisk notation (*) can be entered in the command line to repeat the value in the last line of the previous page until the end of the current page. For attribute control variables, a screen is displayed where you can select attributes and colors as initial values. For details on attributes and colors, see the session parameters AD and CD in the Natural Parameter Reference documentation.
F	This function, too, enables you to define an initial value for the specified field. However, a free-mode editor is provided where you can enter your initial value definitions according to the common Natural syntax definitions. In this way, for example, the same initial value can be assigned to a whole range of field occurrences at a time. During editing, however, the specified values are not checked (unless you enter the CHECK command).
E	This function enables you to define an edit mask and/or header for the specified field, according to the Natural rules for edit mask and header specification. If both an edit mask and a header are specified, together they must not exceed 57 characters in length. However, if only an edit mask is specified, it can be up to 63 characters long; if only a header has been specified, it can be up to 58 characters long. If ".E" is entered for a DDM field, this function is invoked immediately, as only edit masks and headers can be defined for DDM fields. It is not possible to define initial values for DDM fields.
P	This function only applies to parameter data areas and enables you to specify a parameter BY REFERENCE (default), BY VALUE or BY VALUE RESULT, and OPTIONAL. See also Parameter-Data-Definition in the DEFINE DATA section of the Natural Statements documentation.
A	This function enables you to define array bounds for the specified field. A free-mode editor is provided where you can enter your bound definitions in accordance with the common Natural syntax definitions. While you are editing, however, the specified values will not be checked (unless you enter the CHECK command).
D	This function enables you to delete, at a stroke, all definitions made via the "S", "F" and "E" functions. Any "yes" status messages are changed to "no".

Any definitions made within the Initial Values and Edit Mask/Header subfunctions are immediately incorporated into the data area currently displayed in the data area editor.

For an attribute control variable, only the function codes S, F and A are allowed.

For a field that redefines another field, only the function code A is allowed.

After any of the above listed function codes (except function code D) has been entered on the Extended Field Editing screen, a new screen is displayed. There you can enter one of the following commands in the command line:

Command	Function
EDIT	This command returns you to your data area editor screen.
.	This command returns you to the previous screen to continue processing.
- -	This command returns you to the beginning of the initial value specification(s). It is only available for arrays in Single Value Initialization mode.
+	This command takes you one page forward. If the last page has been reached or if there is only one page available, you are returned to your data area editor screen.
*	This command copies the initial value of the last occurrence of the previous page to all empty fields of the current page. It is only available for arrays in Single Value Initialization mode.

Exit Function

If the editor default parameter "Prompt Window for Exit Function" is set to "Y", any time you enter the EXIT command in the command line, the EXIT Function prompt window is invoked, offering you the following options:

Option	Explanation
Save and Exit	Leaves the editor and saves all modifications made to the current object.
Exit without Saving	Leaves the editor without saving any modification made to the current object since the last SAVE command was entered.
Resume Function	Neither leaves the editor nor saves any modifications; the prompt window is closed and the current function is resumed.

When the parameter "Prompt Window for Exit Function" is set to "N", the EXIT command leaves the editor and saves all modifications made to the current object; no prompt window is displayed.

Sharing Data Area Sources by Natural Version 3.1 and 4.1

The data area editor uses an internal layout to store data area sources in the FUSER system file. New features and definitions, that are available with Natural Version 4.1, require that the data area source is stored in the FUSER system file using a new and extended source format.

Data areas that are stored using this new source format cannot be used or edited with Natural Version 3.1. The Natural Version 4.1 data area editor supports the former and the new data area source format. The editor can read both formats and converts the version 3.1 format internally to the new source format. As long as no new features are used, data areas are stored using the former Natural Version 3.1 compatible source format by default to allow for sharing the data area between a Natural Version 3.1 and Natural Version 4.1 environment.

The V31COMP compiler option may be used to ascertain that a data area that is edited and cataloged with the Natural Version 4.1 data area editor can still be edited and cataloged with Natural Version 3.1.

V31COMP = ON

The setting of V31COMP is checked when one of the editor commands save, check, catalog or stow is executed. With check, catalog or stow all differences between Version 4.1 and 3.1 relevant for compilation are checked. See the section Compilation Relevant Differences between Version 4.1 and 3.1. For the save command only the features, which cannot be stored using the former version 3.1 compatible data structure, are not allowed.

The following features cannot be saved in the former internal format:

- The first position of the level number field contains other values than blank or 0.
- The length definition in the length field contains more than 4 bytes. This includes the definition of dynamic variables.
- Array bounds that are defined by using the Array Index Definition function of the Extended Field Definition Editing.
- Definition of optional parameters.

A corresponding message is displayed and you can select to return to the editor or to save the data area in the new source format to avoid the loss of changes.

No error will be reported if V31COMP=ON and the system commands READ and SAVE are executed from the NEXT prompt for a data area containing one of the above mentioned new features.

V31COMP = OFF

By default, the editor will save the data area in the old source format. If new features are used, the editor will generate the new internal format automatically.

Note:
With Natural Version 4.1, the default format for storing data areas in the FUSER system file is the format compatible with Natural Version 3.1. With the next major release of Natural after Version 4.1, the default will be changed to the new and extended format introduced with Natural Version 4.1.

Map Editor

The Natural map editor is used to create maps (screen layouts).

A map can be stored in the Natural system file, from where it can be invoked by a Natural program using an INPUT USING MAP statement (for input maps) or a WRITE USING MAP statement (for output maps).

This section covers the following topics:

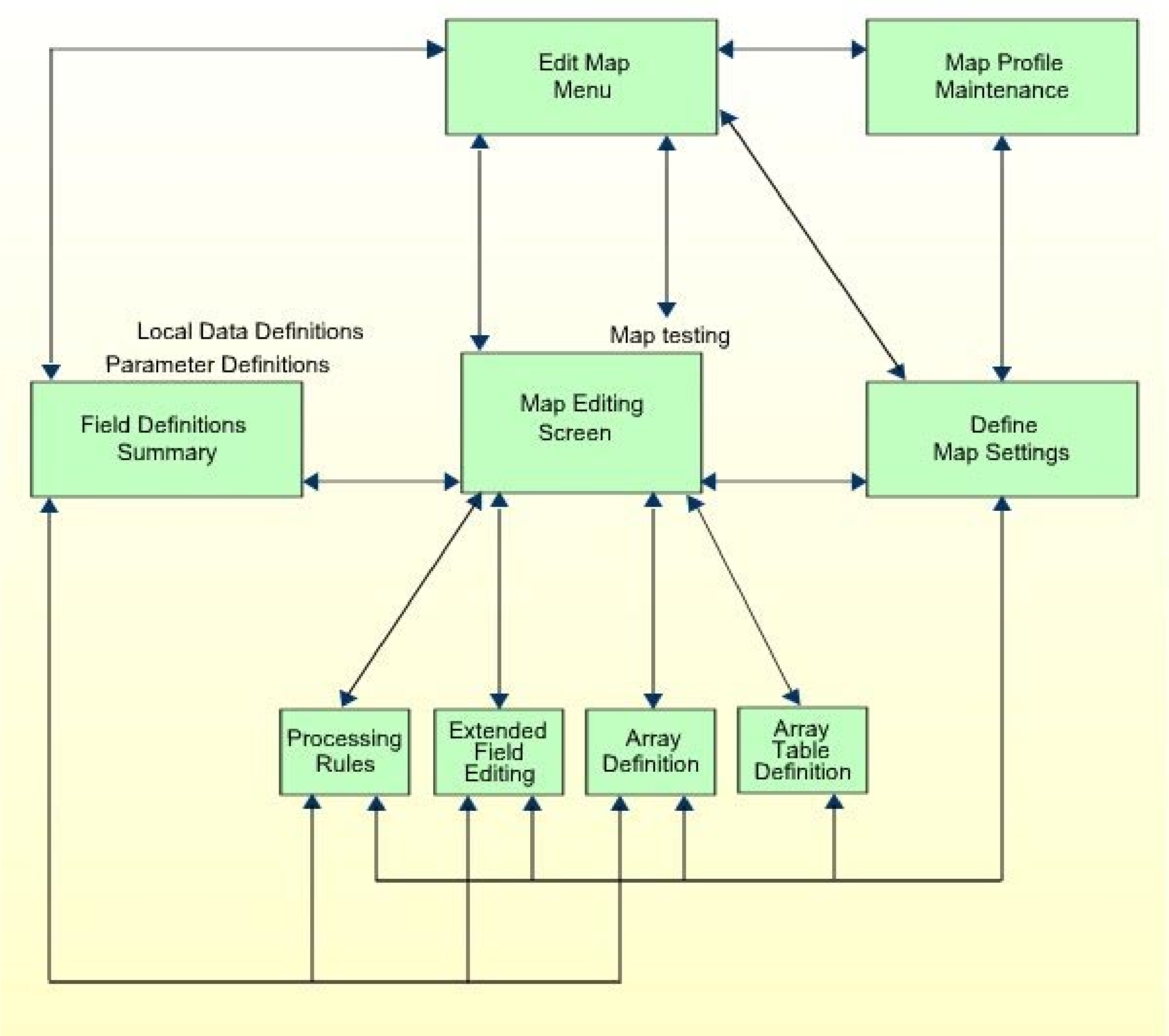
- Components of the Map Editor
- Summary of Map Creation
- Invoking the Map Editor
- Initializing a Map
- Editing a Map
- Defining Map Fields
- Extended Field Editing
- Post Assignment Function
- Array and Table Definition
- Processing Rules

See also:

- Tutorial - Using the Map Editor

Components of the Map Editor

The following figure provides an overview of the various components of the map editor and also shows the possible ways to get from one component to another:



Summary of Map Creation

Map creation involves four major steps:

Step 1

Definition of the map profile (that is, the field delimiters, format settings, context settings and filler characters to be used) by simply selecting the desired settings from a menu.

Step 2

Definition of the map. A map definition can be created before or after the data views that define the fields it contains. These two ways of creating a map definition are:

- First create a prototype map definition, next create the corresponding data views, then integrate the map into the application.
  Fields can be defined directly on the screen. Each field is assigned a default name. Subsequently, when the corresponding data views have been created, the actual field definitions can be assigned to the map fields (post assignment).
- Create a map definition using existing data views.
  If data views already exist, the map fields can be created by using the field definitions contained in the data views. In this case, all characteristics of a field defined in the data views are included when the field is positioned on the screen.

Step 3

Definition of the fields to be used in the map. A full set of map editing facilities is provided which permit simple and efficient map field definition:

- Full-screen or split-screen editing. In split-screen mode, the upper half of the screen is used for the display of user views or data definitions and the lower half for map definition. Map fields can be defined directly on the screen or can be selected from a user view or data definition.
- Screen positioning commands.
- Line commands, which are used to define tables and manipulate lines.
- Field commands, which are used to define arrays and manipulate fields.
- Editor facilities, which are used to edit processing (validation) rules.

Step 4

Storing the map definition. Once created, the map definition can be saved and/or cataloged in the Natural system file. Once saved, a map definition can be read and modified during a subsequent map editor session. Once cataloged, a map definition can be invoked from a Natural program.

Note:
The map editor uses the Auto Save Numbers function of the program and data area editors.

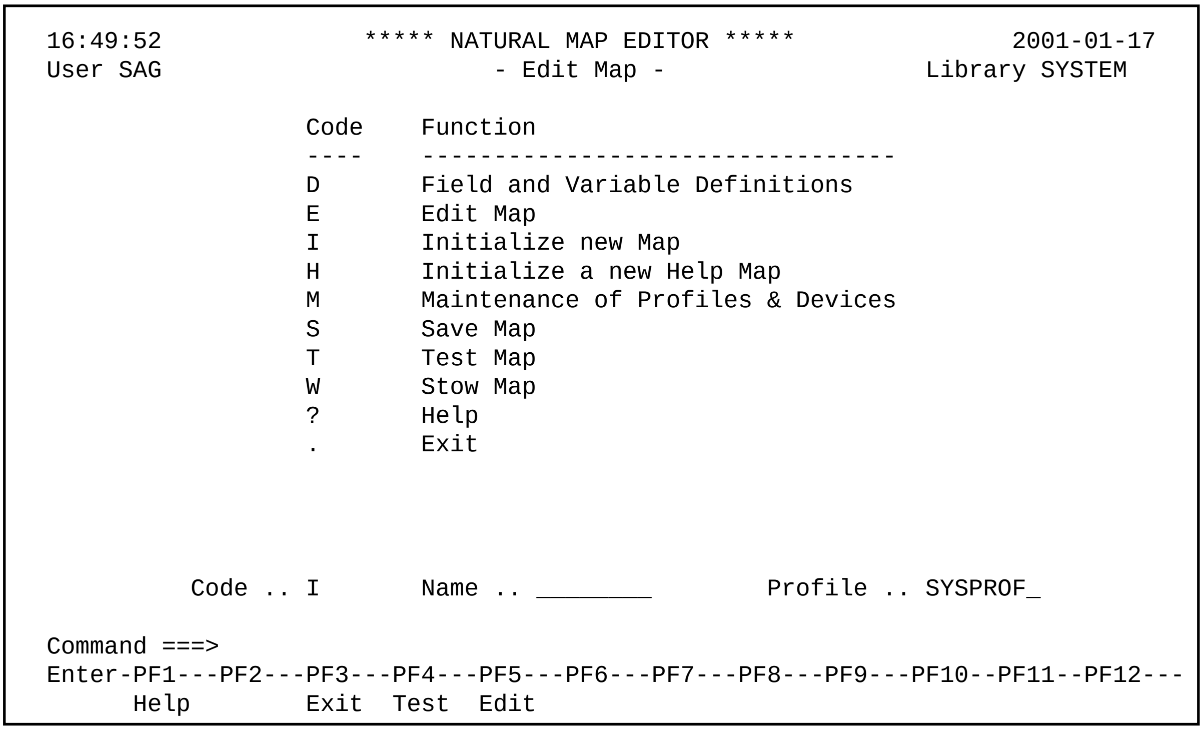
Invoking the Map Editor

You invoke the map editor with the system command:

EDIT MAP

If there is already a map in the source area, the map definition is displayed.

If the source area is empty, the Edit Map menu, which is the main menu of the map editor, is displayed:



```
16:49:52              ***** NATURAL MAP EDITOR *****              2001-01-17
User SAG                       - Edit Map -                     Library SYSTEM

                  Code    Function
                  ----    ---------------------------------
                  D       Field and Variable Definitions
                  E       Edit Map
                  I       Initialize new Map
                  H       Initialize a new Help Map
                  M       Maintenance of Profiles & Devices
                  S       Save Map
                  T       Test Map
                  W       Stow Map
                  ?       Help
                  .       Exit




          Code .. I       Name .. ________        Profile .. SYSPROF_

Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help        Exit  Test  Edit
```

The following entries appear on the Edit Map menu:

Entry	**Explanation**
User	The Natural user ID of the current user.
Library	The Natural library ID currently in effect.
Code	The code of the function to be executed (see below).
Name	The source member which contains the map or help map. For multi-lingual maps, one digit of the source name should be reserved for the language code. For example: USERMAP*1* (language code is *1*) The map above is called from the program by: INPUT USING MAP ’USERMAP&’ where "&" is replaced with the content of the system variable *LANGUAGE at execution time.
Profile	The session profile currently in effect. The profile name is set to the current library ID. If this profile ID is not available, it is set to the current user ID. If this profile ID is not available, the profile name is set to "SYSPROF".

Overview of Functions

The following functions appear on the Edit Map menu:

Field and Variable Definitions

"Field Definitions" displays the following information for each map field:

- Field Name (name of the field)
- Field Mode (type of field), where:
  D means Data Area Field,
  S means System Variable,
  U means User-Defined Field,
  V means View Field,
  blank means Undefined Field
- Field Format (data type and field length)
- Field is an Array (A) or not ("blank")
- Number of attached Processing Rules
- Line and Column position.

This function is equivalent to line command "..E*" entered in the first map line.

The following commands are available within the Field Definitions subfunction:

Command	Description
A	Define array
D	Delete field
E	Edit map field
P*rr*	Edit processing rule
- -	Top
.	Exit

"Variable Definitions" displays all non-map field parameters and all local variables used in the map.

- Name
- Format
- The **Parameter Definitions** function is invoked by pressing PF9 on the Field Definitions screen; new parameters can be added and existing parameters can be modified.
- The **Local Data Definitions** function is invoked by pressing PF10 on the Field Definitions screen; new local variables can be added and existing variables can be modified. Local variables can be used to pass values from one processing rule to another.

The following commands are available within the two Variable Definitions subfunctions:

Command	Description
A	Define array
D	Delete variable
- -	Top
.	Exit

Note:
Command "D" does not delete a parameter if this parameter is still applied to any map field as an attribute control variable, start value or help parameter.

Edit Map

Invokes the map editing screen to modify an existing map or help map definition.

The map editor starts an edit session in split-screen mode, where the upper half of the screen is used for user view definitions and the lower half for map definition. If the map being edited is a help map definition, full-screen mode is in effect.

Initialize a New Map

This function can be executed only if no object with the same name is stored in the Natural system file.

Initialize a New Help Map

This function should be used to create a help map, since it offers you the most flexibility when entering and editing text (leading blanks must be entered). It also provides additional checks to ensure that a valid help map is created.

The function can be executed only if no source and no object with the same name is present in the Natural system file.

A help map is stored as a map and can be referenced with the parameter "HE" in the map definition.

When initializing or editing a help map, you can specify in the map settings where the help map is to appear on the screen at execution time.

Maintenance of Profiles & Devices

This function allows you to add, modify or delete session, map and device profiles.

A session profile is used to assign default map settings to be used when a map or a help map is initialized.

A map profile defines the map settings to be in effect during map definition and execution.

A device profile defines the standard characteristics and settings for a device. This profile can be used to ensure compatibility between the map definition and the device to be used.

See also the section Context and setting Device Check.

Save Map

The map definition is stored in source form in the Natural system file.

Test Map

The current map definition is tested to ensure that it can be executed successfully. This includes testing of all processing rules and help facilities.

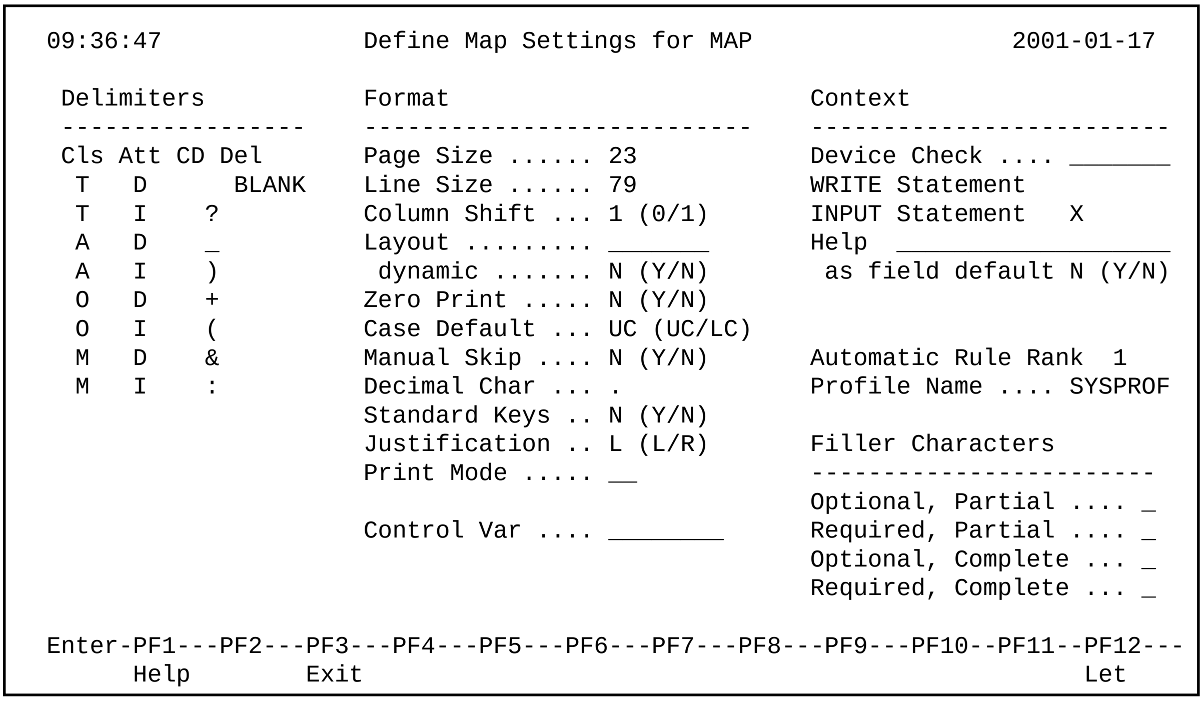
When testing a map, any additionally created numeric map parameters are initialized with the value 1.

Stow Map

Catalog (and save) a map definition. The map definition is cataloged and also stored in source form in the Natural system file.

Initializing a Map

This section describes the process of defining the map settings (profile) for a map or help map definition. When you select the function "Initialize New Map" or "Initialize a New Help Map", the first screen to be invoked is the Define Map Settings screen:



```
09:36:47              Define Map Settings for MAP                  2001-01-17

 Delimiters           Format                         Context
 -----------------    ---------------------------    -------------------------
 Cls Att CD Del       Page Size ...... 23            Device Check .... _______
  T   D      BLANK    Line Size ...... 79            WRITE Statement
  T   I    ?          Column Shift ... 1 (0/1)       INPUT Statement   X
  A   D    _          Layout ......... _______       Help  ___________________
  A   I    )           dynamic ....... N (Y/N)        as field default N (Y/N)
  O   D    +          Zero Print ..... N (Y/N)
  O   I    (          Case Default ... UC (UC/LC)
  M   D    &          Manual Skip .... N (Y/N)       Automatic Rule Rank  1
  M   I    :          Decimal Char ... .             Profile Name .... SYSPROF
                      Standard Keys .. N (Y/N)
                      Justification .. L (L/R)       Filler Characters
                      Print Mode ..... __            ------------------------
                                                     Optional, Partial .... _
                      Control Var .... ________      Required, Partial .... _
                                                     Optional, Complete ... _
                                                     Required, Complete ... _

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help        Exit                                                  Let
```

The Define Map Settings screen comprises the sections:

- Delimiters
- Format
- Context
- Filler Characters

Delimiters

Delimiters are used as a prefix to a field or a text constant to indicate class, attribute and color to be assigned to the field or text constant.

Valid classes are:

Class	Description
A	Input field
M	Output field which is modifiable
O	Output field which is not modifiable
T	Text constant

Valid attributes are:

Attribute	Description
B	Blinking
C	Cursive/italic
D	Default (non-intensified, non-blinking, etc.)
I	Intensified
N	Non-display
U	Underlined
V	Reversed video
Y	Dynamic (attributes to be assigned dynamically by a program)

Valid colors are:

Abbreviation	Color
BL	Blue
GR	Green
NE	Neutral
PI	Pink
RE	Red
TU	Turquoise
YE	Yellow

Any special character can be defined as a delimiter character - except the control character for terminal commands, the control character for map commands and the decimal notation character.

The *default* delimiter characters and their corresponding class and attribute settings are shown in the following table:

Class (Cls)	Attribute (Attr)	Delimiter (Del)
T (text constant)	D (default)	blank
T (text constant)	I (intensified)	?
A (input only field)	D (default)	_
A (input only field)	I (intensified)	)
O (output only field)	D (default)	+
O (output only field)	I (intensified)	(
M (modifiable field)	D (default)	&
M (modifiable field)	I (intensified)	:

These defaults can be changed by the Natural administrator by creating a session profile SYSPROF. They can be changed by the user by either creating own session profiles or changing map settings during the initialization of the map. This is done by simply entering the desired delimiter value in place of the default assignment.

See the section Defining Map Fields for examples of delimiter usage.

Format

The following map format settings can be used:

Entry	Explanation
Page Size	The number of map lines to be edited (1 - 250); if Standard Keys is set to "Y", the number of lines is restricted to 3 - 250. For a map which is output with a WRITE statement, you specify the number of lines of the logical page output with the WRITE statement, not the map size. Thus, the map can be output several times on one page.
Line Size	The number of map columns to be edited (5 - 249).
Column Shift	Column shift (0 or 1) to be applied to the map. This feature can be used to address all 80 columns on a 80-column screen (Column Shift = 1, Line Size = 80). Positional commands (PF10, PF11) must be used to edit all map positions.
Layout	The name of a map source definition which contains a predefined layout.
dynamic	Y Specifies the layout to be dynamic. The dynamically used layout does not become a fixed part of the map at compilation time, but is executed at runtime. Thus, subsequent modifications of a layout map become effective for all maps using that layout map. If the layout map includes user-defined variables, you have to define these parameters in the map using the layout map. Input fields and modifiable fields in the layout map are not open at runtime. Parameters can be added by pressing PF9 within the Field and Variable Definitions function. N Specifies the layout to be static. The static layout is copied into the source area when a map is initialized. Filler characters are not transferred; "N" is the default setting.
Zero Print	Y Displays a field value of all zeros as one zero only. N Displays a zero value as blanks; "N" is the default setting. This value is copied into the field definition when a new field is created and can be modified for individual fields using the extended field editing function.

Entry	Explanation
Case Default	UC Indicates that all input entered for fields at map execution time is to be converted to upper case, that is, the session parameter AD=T is used as a field default. LC Indicates that no lower to upper case conversion is to be performed, that is, the session parameter AD=W is used as a field default. To make the value LC effective, you have to specify the value ON for the Natural profile parameter LC. This value is copied into the field definition when a new field is created and can be modified for individual fields using the extended field editing function.
Manual Skip	Y Does **not** automatically move the cursor to the next field in the map at execution time even if the current field is completely filled. N Moves the cursor automatically to the next field in the map at execution time when the current field is completely filled; "N" is the default setting.
Decimal Char	The character to be used as the decimal notation character. This character can only be changed with the GLOBALS command.
Standard Keys	Y Leaves the last two lines of the map empty so that function-key specifications can be entered at execution time. N Causes all lines to be used for the map; "N" is the default setting.
Justification	The type of field justification to be used for numeric and alphanumeric fields taken from a user view or data definition: L left justified R right justified This value is copied into the field definition when a new field is created.
Print Mode	The default print mode for variables: C Indicates that an alternative character set is to be used (special character table as defined by the Natural administrator). D Indicates that double byte character mode is to be used. I Indicates inverse print direction. N Indicates standard print direction. This value is copied into the field definition when a new field is created.
Control Var	The name of an attribute control variable, the content of which determines the attribute characteristics of fields and texts that have the attribute definition AD=Y or (Y). The attribute control variable referenced in the map must be defined in the program using that map. Removing an attribute control variable from the format map settings implies that the attribute control variable is removed from the map, too, unless it is associated to any other map field.

Context

The following map context settings can be used:

Entry	Explanation
Device Check	If a device name is entered in this field, the map settings are checked for compatibility with the device profile of the specified device. If a setting is not compatible, a warning message is issued (see also the section Maintenance of Profiles & Devices).
WRITE Statement	Marking this field with a non-blank value produces a WRITE statement at the end of the map definition process. The resulting map can then be invoked from a Natural program using a WRITE USING MAP statement. Empty lines at the end of the map are automatically deleted so that the map can be output several times on one page.
INPUT Statement	Marking this field with a non-blank value causes the result of the map definition process to be an INPUT statement. The resulting map can then be invoked from a Natural program using an INPUT USING MAP statement.
Help	The name of a helproutine which is invoked at execution time when the help function is invoked for this map (global help for map). For a detailed explanation of the syntax, refer to the specification of the parameter "HE".
as field default	Y Specifies that the helproutine for the map is to apply as default to each individual field on the map, which means that the name of each field is passed individually to the helproutine. N Specifies that the name of the map is passed to the helproutine; "N" is the default setting. **Note:** If you define the map settings for a help map, on the Define Map Settings for HELPMAP screen, the "Help" and "as field defaults" fields are replaced by the "Position Line Col" field described below.
Position Line Col	The position where the help map is to appear on the screen at execution time. This field only appears if you define the map settings for a help map. It replaces the "Help" and "as field defaults" fields on the Define Map Settings for HELPMAP screen.
Automatic Rule Rank	The rank (priority) assigned to automatic Predict rules when they are linked to the map during field definition. Default is 1.
Profile Name	The name of the profile which was active at map initialization time. If "ENFORCED" is displayed, the following map settings are protected: ● all map delimiters ● static and dynamic layout ● device check ● WRITE and INPUT statements ● all filler characters ● automatic rule rank ● positioning of help maps The name of the profile active at the time the map is created is stored within the map. When the map is edited later and another profile is active, a warning is produced but editing is allowed.

Filler Characters

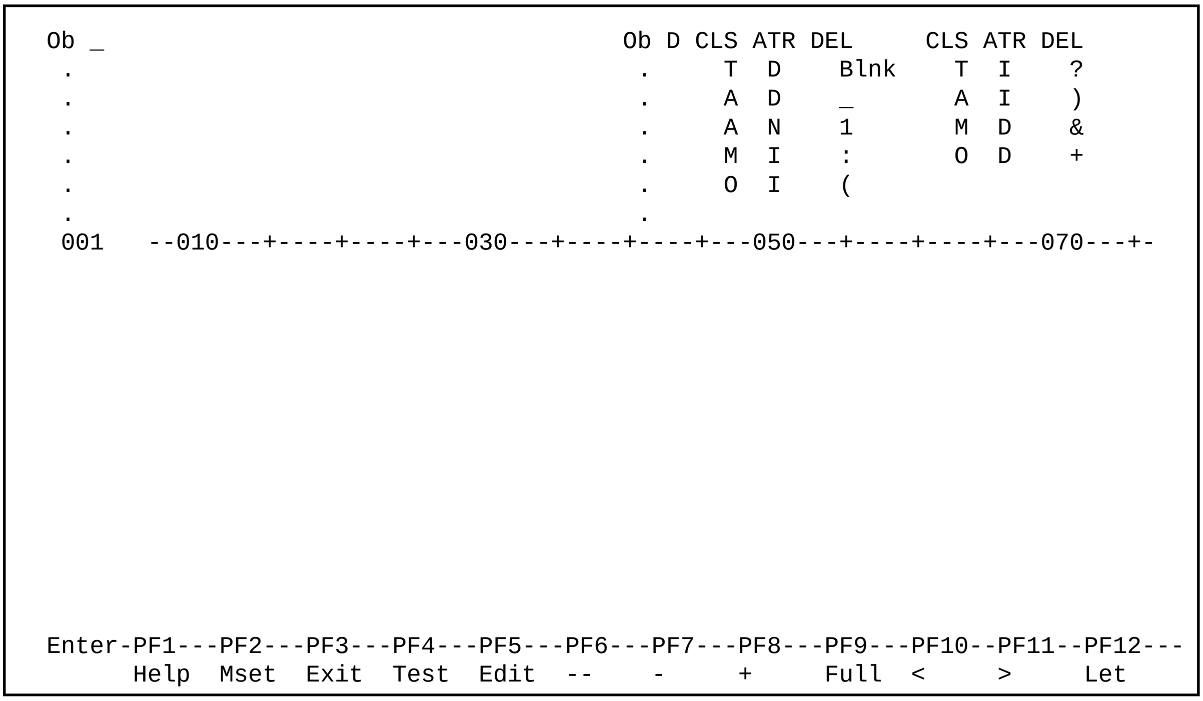
Filler characters can be assigned to indicate whether information for a field is mandatory and whether the field must be completely filled:

Field Type	Explanation
Optional, Partial	Input not mandatory, field need not be completely filled.
Required, Partial	Input mandatory, field need not be completely filled (AD=E).
Optional, Complete	Input not mandatory; if filled, field must be completely filled (AD=G).
Required, Complete	Input mandatory, field must be completely filled (AD=EG).

Filler characters can also be defined for individual fields using the extended field editing function. For definition of field types, see also the session parameter AD as described in the Natural Parameter Reference documentation.

Editing a Map

The map editor begins an edit session always in split-screen mode, which means that the upper half of the screen is used for user-view definitions and the lower half for map definition:



```
Ob _                                    Ob D CLS ATR DEL     CLS ATR DEL
 .                                       .     T  D    Blnk    T  I    ?
 .                                       .     A  D    _       A  I    )
 .                                       .     A  N    1       M  D    &
 .                                       .     M  I    :       O  D    +
 .                                       .     O  I    (
 .                                       .
 001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+-











Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test  Edit  --    -     +     Full  <     >     Let
```

PF9 can be used to switch between full-screen and split-screen mode.

If the rightmost of the view definition windows does not contain a view, the current delimiter settings are displayed in this window instead.

Entering a period (.) in the first position of the leftmost view window returns you to the Edit Map menu.

The following commands and functions are available for editing a map:

- Commands and Function Keys for Positioning
- Line Commands
- Field Commands

Commands and Function Keys for Positioning

The commands and PF keys listed below can be used for map positioning on the screen; you enter the commands at the beginning of a map line:

Key	Command	Function
PF1		Invoke map editor help facility.
PF2		Display/modify the current map settings.
PF3	.Q	Terminate map editing and return to Edit Map menu.
PF4		Test the map definition (without Predict rules).
PF5		Invoke extended field editing for field at which the cursor is currently positioned.
PF6	.- -	Move to top of map.
PF7	.-	Move upwards half a window page.
	.-*nnn*	Move upwards *nnn* lines.
PF8	.+	Move downwards half a window page.
	.+*nnn*	Move downwards *nnn* lines.
	.++	Move to bottom of map.
PF9	./	Switch between split-screen and full-screen mode.
PF10	.<	Move to the left half a window page.
	.<*nnn*	Move to the left *nnn* columns.
	.<<	Move to the left border of the map.
PF11	.>	Move to the right half a window page.
	.>*nnn*	Move to the right *nnn* columns.
	.>>	Move to the right border of the map.
PF12		Ignore changes made on screen subsequent to last use of ENTER.
	.*	Move top left corner to cursor position.

Line Commands

Line commands must be entered in the form "***..line-command***" where ".." represents two occurrences of the control character in effect for the map definition.

It is recommended that you enter a blank at the end of each line command. This prevents the editor from attempting to interpret any information existing on the line as part of the line command.

The following line commands are available:

Command	Function
..A	Array table definition.
..A*n*	Array table definition with *n* occurrences. This command can be used to create a table with *n* occurrences vertically for all fields specified in the current line.
..C	Center a single line (that is, the line in which the command was entered). Two "..C" commands entered on the same screen center the first line and adjust the rest of the selected lines.
..C*n*	Center line and move the *n*-1 lines below it accordingly.
..C*	Center line and move all lines below it accordingly.
..D	Delete a single line (that is, the line in which the command was entered). Two "..D" commands entered on the same screen delete the block of lines delimited by these commands.
..D*n*	Delete line and the *n*-1 lines below it.
..D*	Delete line and all lines below it. If the delete operation affects array elements the array is deleted in total.
..E	Invoke the extended field editing function for all fields contained in the line. Two "..E" commands entered on the same screen display all fields within the range of lines delimited by these commands for possible extended field editing.
..E*n*	Invoke extended field editing for the line and the *n*-1 lines below it.
..E*	Invoke extended field editing for the line and all lines below it. The "..E" commands display a screen with the name and format of the requested fields. The field names shown can be modified. The CMD column can be used to select the desired function: extended field editing, array definition and processing rule editing.
..F*c*	Fill the empty spaces of a line with the character *c*.
..I	Insert a single line. The last empty line on the screen is deleted in order to allow for the line insertion.
..I*n*	Insert *n* lines below the line in which the command was entered.
..I*	Insert as many lines as possible below the command line.

Command	Function
..J	Join the line in which the command was entered with the line below it. Two "..J" commands entered on the same screen joins the range of lines delimited by the commands.
..J*n*	Join the line in which the command was entered with the *n*-1 lines below it.
..J*	Join the line with all lines below it. If a join operation results in a line being too long, the lower line is split at the rightmost possible position and the left part is then joined with the previous line. The right part of the split line is then shifted to the left to align it with the line in which the command was entered.
..M	Move the line in which the command was entered below the cursor line. If two "..M" commands are entered on the same screen, the block of lines delimited by the commands is moved below the line marked with the cursor.
..M*n*	Move the line and the *n*-1 lines below it below the line marked with the cursor.
..M*	Move the line in which the command is entered and all lines below it to the line below the line marked with the cursor. This command is only practical if the line marked with the cursor is above the line in which the command is entered.
..P	Invoke PF-key processing rule editing. PF-key processing rules are special processing rules to define activities assigned to program sensitive function keys.
..P*n*	Invoke PF-key processing rule editing for rank level *n*.
..Q	Terminate map editing and return to the Edit Map menu.
..R	Repeat once all text constants on the line in which the command was entered. The cursor position is used to indicate the target line. If two "..R" commands are entered on the same screen, the text constants within the block of lines delimited by the commands are repeated.
..R*n*	Repeat all text constants on this and the *n*-1 following lines. If the cursor is located below the command line, the same text is repeated *n* times.
..S	Split line at cursor position. If two "..S" commands are entered on the same screen, the block of lines delimited by the commands are split.
..S*n*	Split the line where the command is entered and the *n*-1 lines below it at the cursor position.

Field Commands

Field commands must be entered in the form ".*field-command*" where "." represents the control character in effect for the map definition. Each command must begin in the first position of a map field or text constant.

A field command can be applied to a range of fields or constants. A range can be specified in any of the following ways:

- Two or more of the same field commands can be used on the same screen. The column range (horizontal range) and the line range (vertical range) are determined by the positions of the commands. (The section Tutorial - Using the Map Editor provides examples which illustrate this.)
- A repetition factor *n* can be used. It can be enclosed within parentheses. The command is applied to the designated field and also to the fields in the *n*-1 lines below it. A repetition factor of "*" causes repetition until the bottom of the map is reached.

It is recommended that you enter a blank at the end of each field command. This prevents the editor from attempting to interpret part of the field as part of the field command.

The following field commands are available:

Command	Function
.A	Define an Array. This command can be applied to a single field only and not to a range of fields. The array definition is specified on the screen provided. The resulting array is positioned with its left upper corner at the position where this command was entered. An array can be redefined by applying the ".A" command to one of its elements.
.A*n*	Supply a repetition factor *n* with the ".A" command for the purpose of defining a one dimensional array (no spacing, no offsets) without having to use a separate screen.
.C	Center a field or range of fields between adjoining fields. To center a single field, enter ".C" in the field to be centered. To center a range of fields, enter ".C" in the first and last field to be centered, or enter ".C" in the first field and position the cursor to the last field to be centered. In the event that an adjoining field or fields are not present, the column boundaries in effect for the map definition are used instead.
.D	Delete a field or range of fields. To delete a single field, enter ".D" in the field to be deleted. To delete a range of fields, enter ".D" in the first and last field to be deleted. The field range to be deleted may extend beyond a single line. If an array element is deleted, the entire array is deleted.
.E	Invoke extended field editing for a field. This command can be applied only to a single field and not to a range of fields. Extended field editing can also be invoked by positioning the cursor to the selected field and pressing PF5.
.J	Join fields located on consecutive lines. The left boundary of the join operation corresponds to where the ".J" command is entered and the right one corresponds to the cursor position.

Command	Function
.M	Move a field or range of fields. To move a single field, enter ".M" in the field to be moved and place the cursor at the target position. To move a range of fields, enter ".M" in the first and last field to be moved and place the cursor at the target position.
.P[*n*]	Edit processing rules for a field. Supply a parameter *n* with the ".P" command to indicate the priority (rank) of the processing rule to be edited. If necessary, the value specified for *n* can be included in parentheses "()".
.R	Repeat (copy) a field or range of fields. To copy a single field, enter ".R" in the field to be copied and place the cursor at the target position. To copy a range of fields, enter ".R" in the first and last field to be copied and place the cursor at the target position. Repetition is always done downwards and from left to right. Fields generated by this command are assigned a dummy name. A valid name for each such field must be defined by using the post assignment function or the extended field editing function. **Note:** Arrays cannot be copied.
.S	Split (move) a line or a line range. Enter ".S" in the field at which splitting is to begin and place the cursor at the target position. The line is divided at the position where the ".S" command was entered. The right portion is then moved to the cursor position.
.T	Truncate (delete) a field or range of fields from a line. Enter ".T" in the field at which truncation is to begin. If this function is used to truncate (delete) an array element, the entire array is deleted.

Defining Map Fields

The fields which are to comprise a map definition can be specified in any of the following ways:

- Defining Fields Directly on the Screen
- Selecting Fields from a User View or Data Definition
- Using System Variables in a Map Definition

Defining Fields Directly on the Screen

The fields which are to comprise the map definition are specified by entering a delimiter character followed by the number of positions to be allocated for the field. The following characters can be used:

Character	Meaning
9	Numeric position
0	Numeric right justified
.	Decimal notation (numeric field only)
S	Sign position (numeric field only)
HH	Hexadecimal (binary) (must be entered in groups of two)
X	Alphanumeric position

A repetition factor can also be specified in the form (*n*), for example, "X(5)" is equivalent to "XXXXX".

The following are examples of field definitions (the delimiter character can be changed as desired).

:999	3 positions, numeric
:000	3 positions, numeric right justified
:99.9	3 positions numeric with decimal point
:S9(6)	6 positions, signed numeric
:HHHH	4 positions, hexadecimal
:X	1 position, alphanumeric
:X(7)	7 positions, alphanumeric

Fields entered as shown above are assigned a dummy field name by the map editor. Each field must be assigned a name prior to map execution by using either the extended field editing or post assignment function. Other field formats can be specified using extended field editing.

Selecting Fields from a User View or Data Definition

A field can be selected from a user view or a data definition. The user view or data definition must first be specified next to the entry "Ob:" (object) on the screen (a second user view can also be specified on the right side of the screen).

To select a user view or data definition, first specify the object class and then the object name. Valid object classes are:

Class	Description
A	Parameter Data Area
C	Predict Conceptual Files (only if Predict is installed)
G	Global Data Area
H	Helproutine
L	Local Data Area
M	Map
N	Subprogram
P	Program
S	Subroutine
V	View

Programs, subroutines, subprograms and helproutines can only be used if they contain a DEFINE DATA statement. Data areas should only be used if they are STOWed.

Once a user view has been selected, it can be positioned forwards or backwards on the screen using positioning commands (+,-,++,--,+*n*,-*n*).

To include a user view field in the map definition, enter a delimiter character followed by the number (left-side view) or letter (right-side view) of the desired field. A group or items preceded by a period cannot be selected:

:3 (field 3 of the left-side view is selected)
:C (field C of the right-side view is selected)

Once all user view fields have been selected, press ENTER to show the fields on the map definition. If a selected field contains an edit mask, this is denoted by the notation "M".

The user view field name is used as the map field name for fields selected from a user view.

Using System Variables in a Map Definition

Natural system variables can also be specified in a map definition. The Natural Programming Guide contains a complete description of Natural system variables.

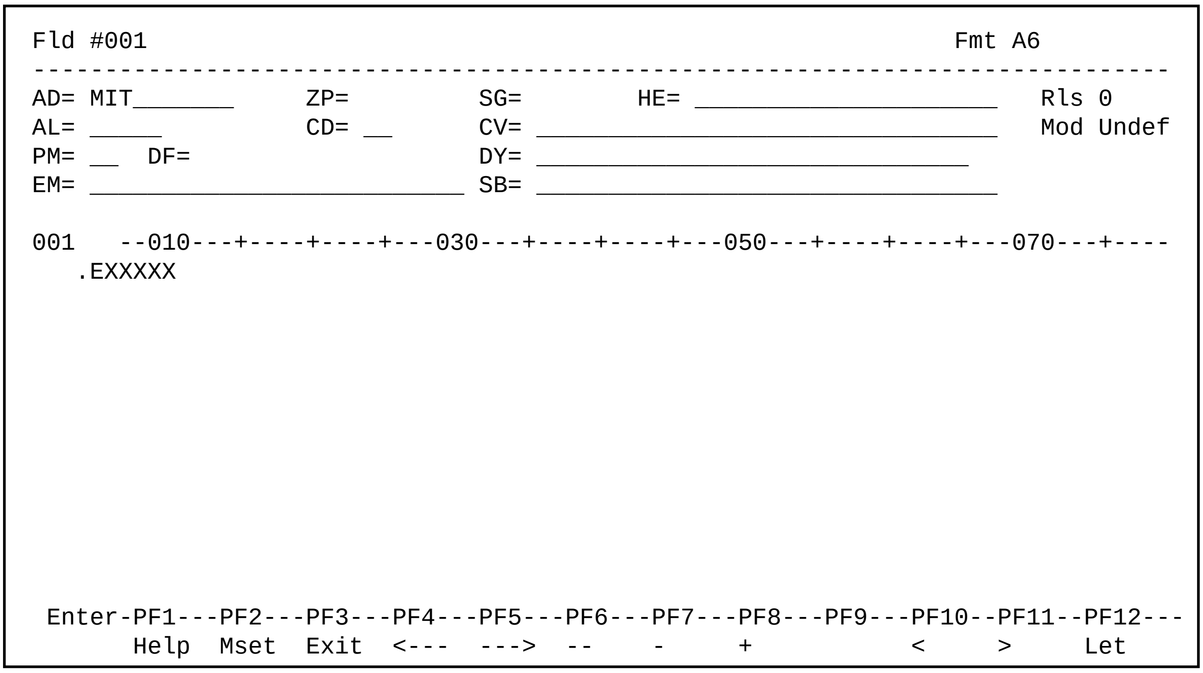
A system variable must be preceded by an output delimiter:

(*TIME
(*DATE
(*APPLIC-ID

Extended Field Editing

Extended field editing is used to define field attributes.

Extended field editing is invoked by entering the line command "..E" or the field command ".E"; the following screen is displayed:



```
Fld #001                                                        Fmt A6
-------------------------------------------------------------------------------
AD= MIT_______     ZP=         SG=        HE= _____________________   Rls 0
AL= _____          CD= __      CV= ________________________________   Mod Undef
PM= __  DF=                    DY= ______________________________
EM= __________________________ SB= ________________________________

001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
   .EXXXXX









 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help  Mset  Exit  <---  --->  --    -     +           <     >     Let
```

It is possible to invoke extended field editing for the next or previous field in the map by pressing PF4 or PF5 respectively, or to invoke extended field editing for any field in the map by moving the cursor onto the desired field and pressing ENTER.

The sample screen above contains the following entries:

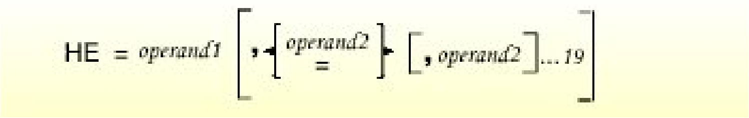
Entry	Explanation
Fld	The field or array name. If the name is longer than the available space, enter **.e** at the beginning of the line to open additional space. Field name assignment is related to the method with which the field was originally defined. If the field was taken from a user view or data definition, it is assigned the same name as the field in the user view or data definition. If the field was specified as a Natural system variable, it is assigned the name of the specified variable. If the field is neither of the above, it is assigned a dummy name. You must assign a name to such a field prior to map execution. The name of a field can be changed. However, a prefix cannot be used for a field which did not have a prefix assigned previously. To obtain a prefixed field name, select the field from a user view or data definition. **Note:** Duplicate field names are only allowed for fields defined as "output only fields". See the section Defining Map Fields for additional information.
Arr	
Fmt	The format and length of the field. These can be changed by overwriting the current entry. To define a reference to a dynamic alphanumeric variable, specify "(D)" or "dynamic" behind the entry. The AL parameter will automatically be set either to the specified value or the maximum length available on the screen.
AL/FL/NL	The length to be used when displaying the field. For dynamic variables and long variables the length is automatically set, but can be modified.
Rls	The number of processing rules currently defined for the field.
ZP	Zero printing. OFF indicates that zero values for the field are **not** to be printed. ON indicates that zero values are to be printed. ZP appears on the screen only if the field is numeric.
SG	Sign position for numeric fields. OFF indicates that **no** sign position is to be allocated (default). ON indicates that a sign position is to be allocated. SG appears on the screen only if the field is numeric.

Entry	Explanation
PM	Print Mode. C indicates that an alternative character set is to be used (as defined by the Natural administrator). D indicates that double-byte character set is to be used. I indicates inverse print direction. N indicates that it is not possible to print a hardcopy of the field content.
DF	Date format (applies only to date fields): Determines the length of a date when converted to alphanumeric representation without an edit mask being specified: S 8-byte representation with 2-digit year component and delimiters (*yy*-*mm*-*dd*). I 8-byte representation with 4-digit year component and no delimiters (*yyyymmdd*). L 10-byte representation with 4-digit year component and delimiters (*yyyy*-*mm*-*dd*). For further information, see the session parameter DF as described in the Natural Parameter Reference documentation.
DY	Dynamic string attributes. The dynamic string parameter is used to define certain characters contained in the text string of an alphanumeric variable to control the attribute setting. See also the session parameter DY as described in the Natural Parameter Reference documentation.

Entry	Explanation
HE	The name of a helproutine to be assigned to the field. For the syntax of the HE parameter, see below. For a detailed explanation of the operands used in the HE option, see the session parameter HE as described in the Natural Parameter Reference documentation. *Operand1* can be the name of a helproutine specified in single quotes (’) or a variable name. ● If a field with the name specified as *operand1* in the HE option exists as a field of a map, the parameter references this field. ● If no field with that name exists in the map, the parameter is defined as A8 (default format assumed) in the map. The format/length of *operand2* is defined in the following way: ● If the parameter specified as *operand2* in the HE option is defined as a field of a map, the parameter references this field. ● If no field with that name exists, the parameter is defined as N7 (default format assumed) in the map. Removing a parameter from the HE option implies that the parameter is also removed from the map, unless it is a map field or it is associated with any other map field as a help parameter or "Starting from" value. Non-map field parameters can be edited in the Field and Variable Definitions screen using PF9. Entering "HE=+" or ".e" opens a window, which provides sufficient space for specifying multiple parameters to be passed to a helproutine.
AD	Field attributes. For source optimization reasons, the default values "D", "H", "F" and "W" are accepted but not retained (see also session parameter AD as described in the Natural Parameter Reference documentation).
CD	Color attributes.

Entry	Explanation
CV	Attribute control variable for dynamic field attributes. The name of a variable which contains the attributes to be used for this field. This variable must be defined with format C in the program. The attribute control variable also contains a "MODIFIED" status indication which indicates whether the field has been modified following map execution. A single attribute control variable can be applied to several map fields, in which case the "MODIFIED" status indication is set if any of the fields referencing the control variable has been modified. To check if an attribute control variable has been assigned the status "MODIFIED", use the MODIFIED option. The attribute control variable can be expanded up to three dimensions, for example, CONTR(*), CONTR(*,*), CONTR(*,*,*), depending on the rank of the corresponding array. **Note:** Removing an attribute control variable from a field implies that the attribute control variable is removed from the map, too, unless it is associated to any other map field.
EM	Edit mask to be used for the field. If the mask exceeds the available space, enter **.e** at the beginning of the line to open additional space.
SB	The name of an array in which the values for a selection box are provided. The indicator "V" will be displayed to show that a selection box is available. The format of the source field applies. You can change it in the Parameter Definitions window.
MODE	Mode indicates how the field was created: DATA The field was created by selecting a field from a DEFINE DATA definition. SYS The field is a system variable. UNDEF The field was created directly on the screen and has a dummy name. USER The name of the field was created by extended field editing. VIEW The field was created by selecting a field from a view (file).

HE Parameter Syntax:



Operand	Possible Structure					Possible Formats										Referencing Permitted	Dynamic Definition
Operand1	C	S				A										no	no
Operand2	C	S				A										no	no

Post Assignment Function

A field which has been previously defined (in layout) directly on the screen can be assigned the field name and field attributes of a user view field or a DEFINE DATA definition.

Note:
Duplicate field names are only allowed for fields defined as "output-only fields".

A map field which has been created using a DDM field definition can be redefined using the field definition from a view defined in a data area.

Post assignment can be done by entering the user view field number (or letter) as shown in the view window directly behind the delimiter of the field.

This function can only be used if the formats of the layout agrees with the field definition. N and P are considered to be identical numeric.

This function cannot be used for view arrays if one or more dimensions of that array are smaller than the dimensions of the array in the layout.

If a length conflict occurs, an AL/FL/NL attribute is generated to map the field definition to the layout definition with truncation or expansion. Data are truncated when AL/FL/NL is specified.

Array and Table Definition

The array definition function is used to define the occurrences and layout of an array.

Array definition is initiated by the field command ".A" or by issuing the line command "..E" and then marking the desired field with the function code "A".

The table definition function is used to define the occurrences and layout of more than one array at the same time. The arrays must begin in the same map line.

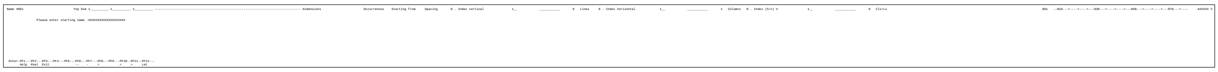
Table definition is invoked by the line command "..A".

Below you will find information on:

- Array Definition
- Table Definition

Array Definition

The upper portion of the following screen is displayed for the purpose of array definition:



```
Name #001              Top Dim 1_________ 1_________ 1_________ ------------------------------------------------------------------------------- Dimensions             Occurrences   Starting from    Spacing      0 . Index vertical           1__            __________       0   Lines     0 . Index horizontal          1__            __________       1   Columns   0 . Index (h/v) V             1__            __________       0   Cls/Ls                                                                                                    001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----     .AXXXXX V
                Please enter starting name .AXXXXXXXXXXXXXXXXXXX




 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help  Mset  Exit                --    -     +           <     >     Let
```

You can specify the following:

Entry	Explanation
Top Dim	Indicates the top dimension of the array; that is, the highest occurrence in (from left to right) the first, second and third dimension. If a field defined in a program is used to define the map array, the upper bounds of that field (user-defined variable or database field), as defined in the program, are used; these cannot be overwritten on the array definition screen. If the map array is derived from a user view array or a data definition, the dimensions of the map array must not exceed the dimensions shown in this field. If the map array is not derived from a user view array or a data definition, the dimensions of the map array must not exceed the dimensions as defined in the Natural program.
Dimensions	An array can have up to three dimensions. The order in which the dimensions of the array are mapped to the map layout is determined by the values entered to the left of the Index operands.
Occurrences	The number of occurrences to be defined for a dimension.
Starting From	The starting index value for a dimension. A numeric value can be used, or a variable name can be used to indicate that the actual value is supplied in the Natural program which invokes the map definition. If the variable is not defined otherwise as a field in the map, it is assumed to be of format/length N7. If so, it can be edited using PF9 in the Field and Parameter Definition screen. **Note:** Removing a "Starting from" value from an array implies that the variable is removed from the map, too, unless it is a map field or it is associated to any other map field as a "Starting from" value or help parameter. To edit "Starting from" values press PF9 in the Field and Variable Definitions screen.
Spacing	The number of blank lines (for vertical dimensions) or blank columns (for horizontal dimensions) to be inserted between each dimension occurrence.

Examples of Array Definitions

Example 1:

A one-dimensional array consisting of 10 vertical occurrences with 2 blank lines to be inserted between each occurrence.

```
Name #001                            Top Dim 10_________ 1__________ 1__________
-------------------------------------------------------------------------------
Dimensions                        Occurrences    Starting from     Spacing
1 . Index vertical                10_            ________          2   Lines
0 . Index horizontal              1__            ________          1   Columns
0 . Index (h/v) V                 1__            ________          0   Cls/Ls
```

Example 2:

Same as example 1 except that the array is to be horizontal.

```
Name #001                            Top Dim 10_________ 1__________ 1__________
-------------------------------------------------------------------------------
Dimensions                        Occurrences    Starting from     Spacing
0 . Index vertical                1__            ________          0   Lines
1 . Index horizontal              10_            ________          1   Columns
0 . Index (h/v) V                 1__            ________          0   Cls/Ls
```

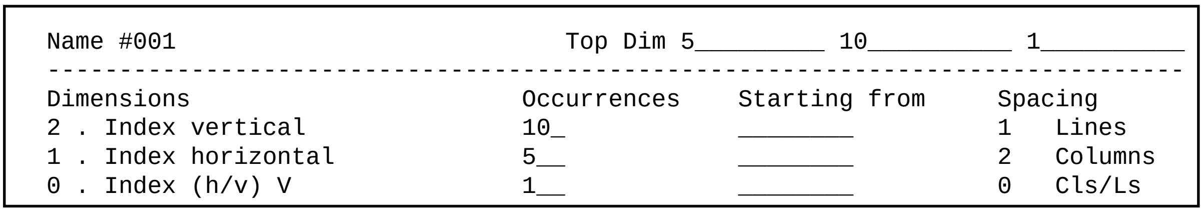
Example 3:

A two-dimensional array. The first dimension consists of 10 vertical occurrences with 1 blank line between each occurrence. The second dimension consists of 5 horizontal occurrences with 2 blank columns between each occurrence.

```
Name #001                            Top Dim 10_________ 5__________ 1__________
-------------------------------------------------------------------------------
Dimensions                        Occurrences    Starting from     Spacing
1 . Index vertical                10_            ________          1   Lines
2 . Index horizontal              5__            ________          2   Columns
0 . Index (h/v) V                 1__            ________          0   Cls/Ls
```

Example 4:

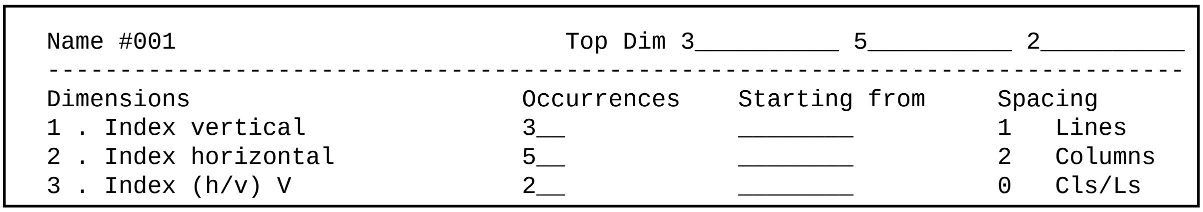
Same as example 3 except that the order of the dimensions is reversed.



```
Name #001                            Top Dim 5_________ 10__________ 1__________
-------------------------------------------------------------------------------
Dimensions                        Occurrences    Starting from     Spacing
2 . Index vertical                10_            ________          1   Lines
1 . Index horizontal              5__            ________          2   Columns
0 . Index (h/v) V                 1__            ________          0   Cls/Ls
```

Example 5:

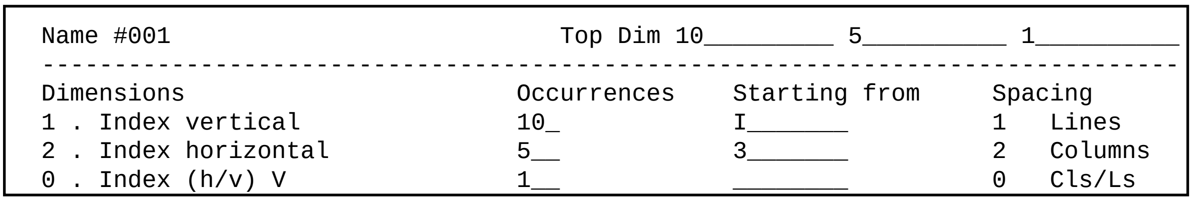
A three-dimensional array. The first dimension consists of 3 vertical occurrences with 1 blank line between each occurrence. The second dimension consists of 5 horizontal occurrences with 2 blank columns between each occurrence. The third dimension consists of 2 occurrences, expanded vertically within each occurrence of the first dimension.



```
Name #001                            Top Dim 3__________ 5__________ 2__________
-------------------------------------------------------------------------------
Dimensions                        Occurrences    Starting from     Spacing
1 . Index vertical                3__            ________          1   Lines
2 . Index horizontal              5__            ________          2   Columns
3 . Index (h/v) V                 2__            ________          0   Cls/Ls
```

Example 6:

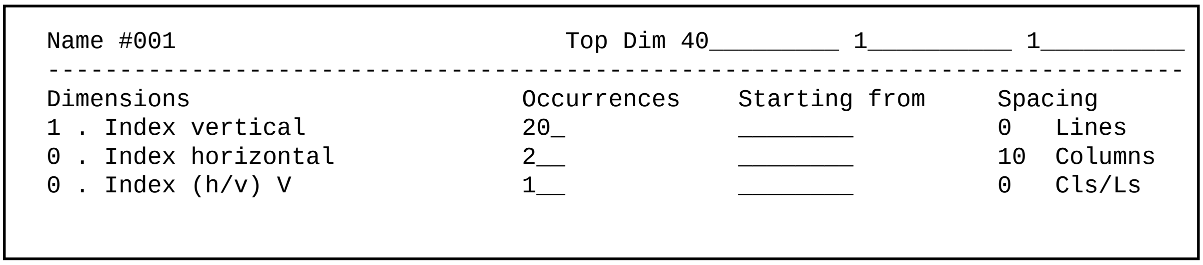
An example using "Starting from". The first dimension consists of 10 vertical occurrences starting from index I. ’I’ is defined in the map editor with format/length N7 by default. The second dimension consists of 5 horizontal occurrences starting from the index 3.



```
Name #001                            Top Dim 10_________ 5__________ 1__________
-------------------------------------------------------------------------------
Dimensions                        Occurrences    Starting from     Spacing
1 . Index vertical                10_            I_______          1   Lines
2 . Index horizontal              5__            3_______          2   Columns
0 . Index (h/v) V                 1__            ________          0   Cls/Ls
```

Example 7:

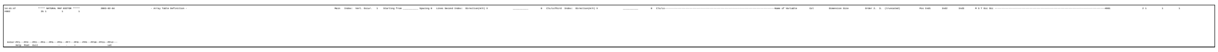
An example of making a two-dimensional display from a one-dimensional array. The array consists of 40 elements. It is displayed in two columns with 20 lines each. This is achieved by specifying 0 as the horizontal index.



```
Name #001                            Top Dim 40_________ 1__________ 1__________
-------------------------------------------------------------------------------
Dimensions                        Occurrences    Starting from     Spacing
1 . Index vertical                20_            ________          0   Lines
0 . Index horizontal              2__            ________          10  Columns
0 . Index (h/v) V                 1__            ________          0   Cls/Ls
```

Table Definition

A table of one or more arrays which all begin in the same map line is defined with the "..A" line command. When you enter the "..A" line command, the following screen is invoked:



```
14:41:47          ***** NATURAL MAP EDITOR *****          2003-02-04          - Array Table Definition -          Main   Index:  Vert. Occur.   1   Starting from __________ Spacing 0   Lines Second Index:  Direction(H/V) V          __________        0   Cls/LsThird  Index:  Direction(H/V) V          __________        0   Cls/Ls--------------------------------------------------------------------------Name of Variable      Col      Dimension Size      Order 2.  3.  (truncated)      Pos Ind1      Ind2      Ind3      M S T Occ Occ ---------------------------------------------------------------------------#001         2 1      1      1
#002                    25 1          1          1



 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help  Mset  Exit              --    -     +                       Let
```

Note:
Applying the "..A" command to arrays which were not defined by an "..A" command but by an ".A" command, may result in a modification or even a destruction of these arrays.

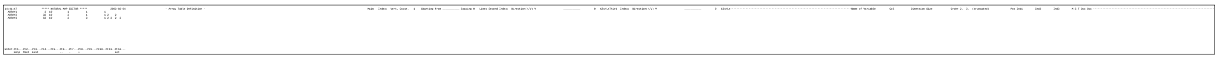
The sample screen above contains the following entries:

Entry	**Explanation**
Main Index	The number of vertical occurrences, the starting position and the number of lines to be skipped between each dimension occurrence.
Second Index	The direction (horizontal or vertical), the starting position and the number of lines/columns to be skipped between each dimension occurrence. The second dimension only applies if one of the arrays has more than one dimension. In this case the second dimension can be displayed either horizontally (in which case there must be enough space in the line for all selected occurrences) or vertically (in which case there must be enough lines on the map to display main dimension times second dimension occurrences, including line spacing).
Third Index	The direction (horizontal or vertical), the starting position and the number of lines/columns to be skipped between each dimension occurrence. The third dimension only applies if one of the arrays has more than two dimensions. In this case the third dimension can be displayed either horizontally (in which case there must be enough space in the line for all selected occurrences) or vertically (in which case there must be enough lines on the map to display main dimension times second dimension times third dimension occurrences, including line spacing).
Name of Variable	All names of field arrays contained in the table are displayed.
Col Pos	The column position in which the field is located. This is displayed for informational purposes only.
Dimension Size	The size of the array as defined in a user view or data definition, or as in a Natural program. If the map array is derived from a user view array or data definition, the dimensions of the map array must not exceed the dimensions shown in this field. If the map array is not derived from a user view array, the dimensions of the map array must not exceed the dimensions as defined in the Natural program.
Order	The order in which the dimensions are to be defined. M, S and T correspond to Main, Second and Third.
2nd Ind Occ.	The number of occurrences to be defined for the second index.
3rd Ind Occ.	The number of occurrences to be defined for the third index.

Example of Table Definition:

```
DEFINE DATA
  1 ARRAY1 (A3/1:10)
  1 ARRAY2 (A5/1:10,1:2)
  1 ARRAY3 (A7/1:10,1:2,1:3)
END-DEFINE
```

Table definition:



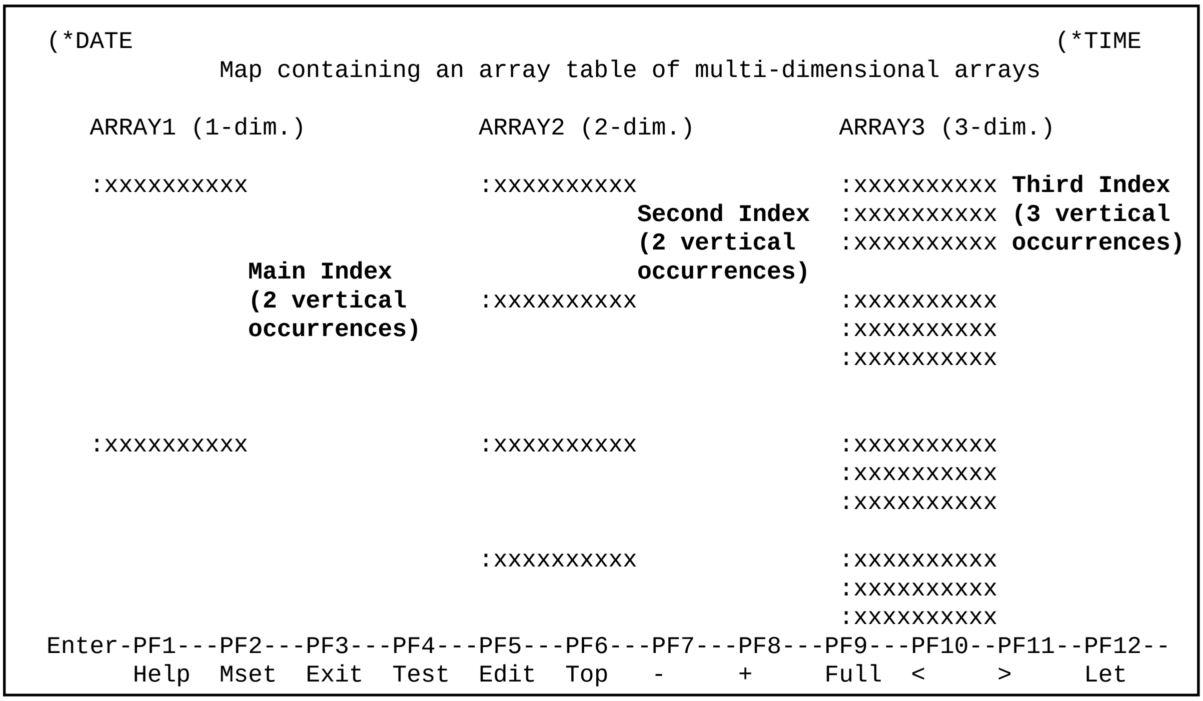
```
14:41:47             ***** NATURAL MAP EDITOR *****             2003-02-04                    - Array Table Definition -                                   Main   Index:  Vert. Occur.   1   Starting from __________ Spacing 0   Lines Second Index:  Direction(H/V) V                __________        0   Cls/LsThird  Index:  Direction(H/V) V                __________        0   Cls/Ls---------------------------------------------------------------------------Name of Variable        Col         Dimension Size          Order 2.  3.  (truncated)            Pos Ind1        Ind2        Ind3        M S T Occ Occ ---------------------------------------------------------------------------
  ARRAY1                 3  10          1           1          1
  ARRAY2                32  10          2           1          1 2    2
  ARRAY3                58  10          2           3          1 2 3  2  3








Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit                --    -     +                       Let
```

ARRAY1 is a one-dimensional array with ten occurrences. The first two occurrences are expanded in the table.

ARRAY2 is a two-dimensional array. The first index consists of ten occurrences and the second index consists of two occurrences. The first two occurrences of the first index and both occurrences of the second index are expanded in the table.

ARRAY3 is a three-dimensional array. The first index consists of ten occurrences, the second index consists of two occurrences and the third index consists of three occurrences. The first two occurrences of the first index, both occurrences of the second index and all three occurrences of the third index are expanded in the table.

Table layout:



```
(*DATE                                                                (*TIME
            Map containing an array table of multi-dimensional arrays

   ARRAY1 (1-dim.)            ARRAY2 (2-dim.)          ARRAY3 (3-dim.)

   :xxxxxxxxxx                :xxxxxxxxxx              :xxxxxxxxxx Third Index
                                         Second Index  :xxxxxxxxxx (3 vertical
                                         (2 vertical   :xxxxxxxxxx occurrences)
              Main Index                 occurrences)
              (2 vertical     :xxxxxxxxxx              :xxxxxxxxxx
              occurrences)                             :xxxxxxxxxx
                                                       :xxxxxxxxxx


   :xxxxxxxxxx                :xxxxxxxxxx              :xxxxxxxxxx
                                                       :xxxxxxxxxx
                                                       :xxxxxxxxxx

                              :xxxxxxxxxx              :xxxxxxxxxx
                                                       :xxxxxxxxxx
                                                       :xxxxxxxxxx
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12--
      Help  Mset  Exit  Test  Edit  Top   -     +     Full  <     >     Let
```

The table is defined as a collection of arrays which share the following characteristics:

- The number of occurrences of the main index must be the same for each array of the table. The main index is always expanded vertically.
- All elements of a specific index must be placed in the same line. Thus, spacing between the elements of a specific index depends on the array with the largest dimension.

Processing Rules

- Field-Related Processing Rules
- Function-Key-Related Processing Rules
- Processing Rule Editing

Field-Related Processing Rules

Three types of processing rules can be defined:

- Inline processing rules
- Free Predict rules
- Automatic Predict rules

Inline processing rules are defined within a map source and do not have a name assigned. The availability of Predict is not required for inline rules.

Free Predict rules have a name assigned and are stored in Predict.

To edit a free processing rule, enter the rule during map creation and assign a name to it. Inline rules can become Predict rules (and vice versa) by assigning/removing the rule name.

Predict automatic rules apply to database fields and are defined by the Predict administrator. When a field is created by selecting it from a view or a data definition, and if the field is a database field, all automatic rules for that field are linked to the map definition. All automatic rules are concatenated and treated as a single map rule.

The rank of the automatic rules is defined in the map settings (default 1).

Automatic rules cannot be modified using the map editor. They can, however, be assigned a different rank either by using the command "P=*n*" or by just overwriting the old rank.

If Predict rules are modified subsequently by the Predict administrator, or new automatic rules are linked to a database field, or automatic rules are removed, it is sufficient to recatalog the map.

> **Note:**
> If a field with linked Predict processing rules is renamed, the rules are lost and must be linked again.

An ampersand "&" within the source code of a processing rule is dynamically substituted with the fully qualified name of the field using the rule; this does not apply if the rule is used by individual array elements.

Example:

```
IF & = ’ ’ REINPUT ’ENTER NAME’ MARK *&
```

The field name notation "**&.***field-name*" within the source code of a processing rule allows you to have DDM-specific rules that cross-check the integrity of values between database fields, without having to explicitly qualify the fields with a view name. As *field-name* you specify the name of the database field as defined in the DDM, and at compilation time, Natural dynamically qualifies the field by replacing the "&" with the corresponding view name. This allows you to use the same processing rule for specific fields, regardless of which view the fields are taken from.

Function-Key-Related Processing Rules

Two types of function-key-related processing rules can be defined:

- Inline processing rules
- Free Predict rules

Function-key-related processing rules can be used to assign activities to program sensitive function keys during map processing. For function keys which already have a command assigned by the program, this command is executed without any rule processing.

Example:

```
IF *PF-KEY = ’PF3’
   ESCAPE ROUTINE
END-IF
```

When this rule is executed, map processing is terminated without further rule processing.

Processing Rule Editing

Processing-rule editing is invoked by the field command ".P", or by issuing the line command "..E" and then placing the function code "P" next to the field for which processing rule editing is to be performed. PF-key processing rule editing is invoked by the command "..P".

A parameter can be used (.P*rr*) to indicate the rank (priority) of the processing rule to be defined/edited. A field can have up to 100 processing rules (rank 0 to 99). At map execution time, the processing rules are executed in ascending order by rank and screen position of the field. PF-key processing rules are always assumed to have the first screen position.

For optimum performance, the following assignments are recommended when assigning ranks to processing rules:

Rank	Processing Rule
0	Termination rule
1 - 4	Automatic rules
5 - 24	Format checking
25 - 44	Value checking for individual fields
45 - 64	Value cross-checking between fields
65 - 84	Database access
85 - 99	Special purpose

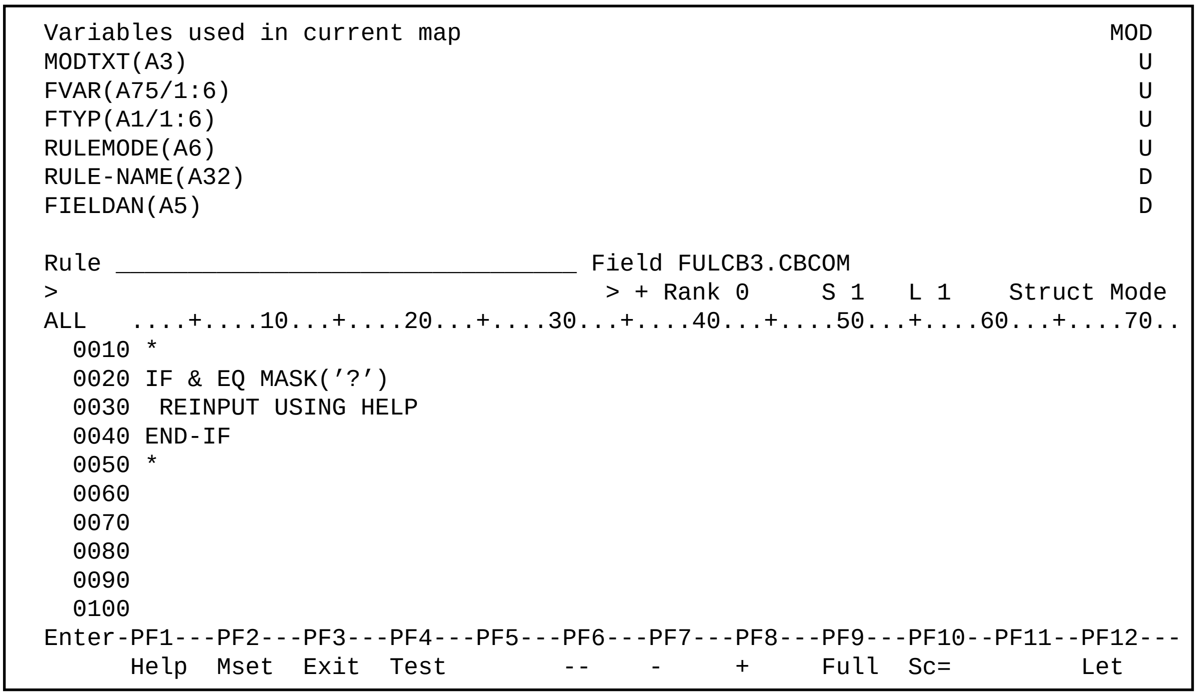
How to Select a Rule for Editing

If you enter the field command ".P*" in a map field, you obtain a list of all processing rules defined for the field.

If you enter the line command "..P*" in any map line, you obtain a list of all function-key-related processing rules defined for the map.

On each list, the Predict rules are identified by their names, the inline rules by their first three source code lines. From each list you can select a rule for editing by entering its rank.

The screen for processing rule editing (with a processing rule example) is shown below:



```
Variables used in current map                                             MOD
MODTXT(A3)                                                                  U
FVAR(A75/1:6)                                                               U
FTYP(A1/1:6)                                                                U
RULEMODE(A6)                                                                U
RULE-NAME(A32)                                                              D
FIELDAN(A5)                                                                 D

Rule ________________________________ Field FULCB3.CBCOM
>                                      > + Rank 0     S 1   L 1    Struct Mode
ALL   ....+....10...+....20...+....30...+....40...+....50...+....60...+....70..
  0010 *
  0020 IF & EQ MASK('?')
  0030  REINPUT USING HELP
  0040 END-IF
  0050 *
  0060
  0070
  0080
  0090
  0100
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test        --    -     +     Full  Sc=         Let
```

During processing rule editing you can switch between split-screen and full-screen mode using PF9 or SPLIT/SPLIT E command. The upper half of the split screen displays the definitions of all map fields (except system variables). This display can be positioned by split-screen commands.

The source code used to define the processing rule is entered/edited in the same way as with the Natural program editor.

While working in the processing-rule editor, processing rules can be edited by entering the following commands in the editor command line:

Command	Function
ADD[(*n*)]	This command adds *n* empty lines in source code. See the more detailed description of the ADD command in the section Editor Commands.
CHANGE ’*string1*’*string2*’	This command scans for the value entered as *string1* and replaces each such value found with the value entered as *string2*.
CHECK	This command checks the rule.
CLEAR	This command clears the edit area (including the line markers "X" and "Y").
DX, DY, DX-Y	This command deletes the X-marked line; or the Y-marked line; or the block of lines delimited by "X" and "Y".
EX, EY, EX-Y	This command deletes source lines from the top of the source area to, but not including, the X-marked line; or from the source line following the Y-marked line to the bottom of the source area; or all source lines in the source area excluding the block of lines delimited by "X" and "Y".
EXIT	This command terminates the rule editing function and return to map editing.
.	
P	This command positions forward to the next rule defined for the field.
P*	This command selects a rule from the selection menu.
P*rr*	This command selects the rule with rank *rr*.
P=*rr*	This command changes the rank of a processing rule to rank *rr*.
POINT	This command positions the line in which the line command ".N" was entered to the top of the current screen.
RESET	This command deletes the current X and/or Y line markers and any marker previously set with the line command ".N".
SAVE *name*	This command saves a rule as copycode with the name *name*.
SCAN [’*scan-value*’]	This command scans for data in the source area. Entering SCAN without any parameter displays the SCAN menu. Entering SCAN ’*scan-value*’ results in a scan for *scan-value*.
SCAN = [+|-]	This command scans for the next occurrence of the scan value. The direction of the scan operation is determined by the setting of the direction indicator. See the more detailed description of the SCAN commands in the section Editor Commands.
SHIFT [-|+*nn*]	This command shifts each source line delimited by the X and Y markers to the left or right. *"nn"* represents the number of characters the source line is to be shifted. Comment lines are not shifted.
SHIFT - -	This command shifts each source line delimited by the X and Y markers to the leftmost position. Comment lines are not shifted.
SHIFT ++	This command shifts each source line delimited by the X and Y markers to the rightmost position (maximum 99 positions). Comment lines are not shifted.

Command	Function
SPLIT [END]	This command switches between split-screen mode and full-screen mode (see also the section Split-Screen Commands).
TEST	This command tests a map.
UNLINK	This command unlinks an inline or Predict free rule from the field.

Note:
To select a rule from all free Predict rules, enter a "?" in the rule name field of the processing rule editing screen.

Editor Commands for Positioning

Editor commands for positioning are also entered in the command line of the rule editor. The following commands are available:

Command	**Function**
+P	Position forwards one page.
+	
-P	Position backwards one page.
-	
+H	Position forwards half a page.
-H	Position backwards half a page.
TOP	Position to top of rule.
- -	
BOTTOM	Position to bottom of rule.
++	
+*nnnn*	Position forwards *nnnn* lines (maximum 4 digits).
-*nnnn*	Position backwards *nnnn* lines (maximum 4 digits).
nnnn	Position to line *nnnn*.
X	Position to the line marked with "X".
Y	Position to the line marked with "Y".
SPLIT[-|+*nn*|++| - -]	Use of positioning commands in split screen.

In split screen mode, all positioning commands must be preceded by an "S" (for Split Screen). See further information in the section Split Screen Commands.

Line Commands

In addition to the editor commands, the following line commands can be used when editing a processing rule:

Command	Function
.C(*nnnn*)	Copies the line in which the command was entered.

.CX(*nnnn*)
.CY(*nnnn*)
	Copies the X-marked or the Y-marked line. See also the commands ".X" and ".Y" in the following section.

.CX-Y(*nnnn*)	Copies the block of lines delimited by the X and Y markers. If the direction indicator is "+", the copied lines are placed after the line in which the command was entered. If the direction indicator is "-", the copied lines are placed before the line in which the command was entered.
.D(*nnnn*)	Delete line or lines. The default is 1 line.
.I(*n*)	Inserts *n* empty lines. With the next ENTER, lines that are left blank are eliminated again.
.I(*obj,ssss,nnnn*)	Inserts an object contained in the current library or in the steplib into the source. The "*ssss*" entry can be used to indicate the line number at which the insert operation is to begin. The "*nnnn*" entry can be used to indicate the number of lines to be inserted. See the more detailed description of the .I line commands in the section Line Commands.
.J	Joins the current line with the next line. If the resulting line length is greater than the length of the editor screen line, the line is marked with "L" and must then be separated again using the ".S" command (see below), before it can be modified.
.L	Undoes all modifications that have been made to the line since the last time ENTER was pressed.
.MX .MY	Moves the X-marked or the Y-marked line. See also the commands ".X" and ".Y" below.
.MX-Y	Moves the block of lines delimited by the X and Y markers. If the direction indicator is set to "+", the moved lines are placed after the line in which the command was entered. If the direction indicator is set to "-", the moved lines are placed before the line in which the command was entered.
.P	Positions the line marked by this command to the top of the screen.
.S	Splits the line at the position marked by the cursor.
.W	Inserts *n* empty lines. With the next ENTER, lines that are left blank are eliminated again.
.X	Marks a line, or the beginning of a block of lines, to be processed.
.Y	Marks a line, or the end of a block of lines, to be processed. **Note:** If both the commands ".X" and ".Y" are applied to one line, it is treated as being marked with "X" and with "Y"; the line marker actually shown to reflect this status is a "Z".